

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
katedra informatiky a výpočetní techniky

DIPLOMOVÁ PRÁCE

**Metody pro porovnání dynamických
trojúhelníkových sítí**

Plzeň 2007

František Zadražil

Metody pro porovnání dynamických trojúhelníkových sítí

Abstrakt Jedním z aktuálních problémů počítačové grafiky je extrakce a zpracování dynamických sítí v reálném čase. Pod pojmem *dynamická síť* chápeme sekvenci trojúhelníkových sítí stejné konektivity, jakožto vývoj modelu v čase. Při řešení této úlohy se potýkáme s obrovským množstvím dat, která je nutno zpracovávat a přenášet po síti. Proto se v poslední době dostává do popředí téma komprese dynamických sítí.

Tato práce se zaměřuje na techniky srovnání dvou dynamických sítí, originální a komprimované. Uvádíme několik metrik, od velmi oblíbené MSE (Mean Squared Error), která je spíše známa pro použití u zpracování obrazu, až po zcela nové metody (*chybové vektory a trojúhelníkové difference*). Zavádíme srovnání výsledků podávaných těmito metrikami, včetně analýzy situací, ve kterých jednotlivé techniky nebudou spolehlivé.

Z hlediska aplikace kompresních algoritmů je také velice důležité, jak případné deformace modelů vnímá člověk. Zabýváme se tedy také subjektivním testováním a analýzou artefaktů, které se mohou u dynamických sítí vyskytnout s ohledem na to, jak je vnímá pozorovatel a výsledky jsou srovnávány s hodnotami objektivního měření metrik.

Methods of triangular dynamic meshes comparison

Abstract Real-time extraction and processing of dynamic meshes is one of the topical issues in computer graphics. By dynamic mesh we mean a sequence of triangular meshes of the same connectivity which defines the progress of the model in time. While solving this problem one stands against the huge amount of data to process and distribute over the network. Hence the subject of dynamic meshes compression has been growing more important recently.

This work focuses on comparison techniques of two dynamic meshes, the original and the compressed one. We mention several metrics, from the widely used MSE (Mean Squared Error), which has been known mainly in the image processing, to completely new methods (*error vectors* and *triangle difference*). We propose the comparison of the result given by these metrics including the analysis of the situations where the techniques will not be reliable.

But the point of view of human observer may be quite different from the objective measure and this point is really important for the application of the compression algorithms. Therefore this work deals with the subjective testing and analysis of artefacts, that can appear in dynamic meshes, with respect to how the observer perceive them. The results are compared with the objective metrics values then.

Poděkování

Rád bych poděkoval Ing. Petru Lobazovi za ochotu a snahu vyjít vstříc. Největší poděkování ale náleží mamince, která mi po celou dobu studia byla velkou oporou.

Datovou sadu *chicken*, která byla použita v této práci, vytvořili Andrew Glassner, Tom McClure, Scott Benza a Mark Van Langeveld. Tato krátká sekvence prostorových dat je šířena výhradně pro účely porovnávání geometrických kompresních technik.

Tato práce byla podporována Ministerstvem školství, mládeže a tělovýchovy České republiky v rámci výzkumného programu LC-06008 (Centrum pro počítačovou grafiku).

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne

František Zadražil

Obsah

1	Úvod	7
1.1	Struktura dokumentu	8
1.2	Pojmy a značení	8
2	Metriky pro porovnání trojúhelníkových sítí	10
2.1	Mean Squared Error	11
2.2	Hausdorffova vzdálenost	12
2.3	KG metrika	15
2.4	Vážené chybové vektory	17
2.5	Trojúhelníková diference	21
2.6	4D Hausdorffova vzdálenost	22
2.7	Ribbon metrika	25
3	Nedostatky metrik	28
3.1	Odchyly v prostoru	29
3.2	Odchyly v čase	31
3.3	Invariance metrik	32
3.3.1	Rotace	33
3.3.2	Změna měřítka	35
3.3.3	Počet snímků	35
3.3.4	Počet bodů	36
4	Výsledky objektivních metod	38
4.1	Implementace	38
4.2	Moduly distorze	39
4.2.1	Gaussovský šum	41

4.2.2	Sinusové prohnutí	41
4.3	Srovnání metrik	43
5	Subjektivní srovnání metrik	45
5.1	Metodika výzkumu	45
5.2	Relevantnost výsledků	46
5.3	Vyhodnocení	46
6	Závěr	54
6.1	Zhodnocení výsledků a budoucí práce	54
A	Dotazník	I
B	Uživatelský manuál	II
B.1	Načtení datové sady	II
B.2	Vykreslení datové sady	III
B.3	Měření odchylky	IV
B.4	Použití 4D metriky	V
C	Obsah přiloženého CD	VII

Kapitola 1

Úvod

Počítačová grafika je v dnešní době jednou z nejrychleji se rozvíjejících oblastí v rámci počítačových technologií. Také uživatelé kladou na grafické výstupy stále větší požadavky. Nové aplikace s sebou přinášejí rostoucí nároky na technické parametry počítačů, které v mnoha případech znamenají velká omezení.

Jedním z příkladů aplikace, kde narážíme na hardwarová omezení, je extrakce a zpracování dynamických sítí v reálném čase. Při jejím řešení se potýkáme s obrovským množstvím dat, která je nutno zpracovávat a přenášet po síti, což je hlavním limitem. Pro statické sítě bylo prezentováno několik přístupů pro snížení jejich složitosti a v několika posledních letech se objevují i techniky, které se snaží redukovat velikost datové reprezentace dynamických sítí 3D modelů (tedy jejich animace), ale přitom zachovat dobrou aproximaci jejich povrchu [Vasa06].

Techniky pro omezení velikosti sítě se dají v podstatě rozdělit do dvou základních kategorií – komprese a zjednodušení. Zatímco komprese se zaměřuje na určité zakódování ukládaných informací a tím o snížení jejich objemu, zjednodušující algoritmy se snaží odstranit ze sítě ty části či detaily, které jsou pro pozorovatele nepodstatné nebo nepostřehnutelné. Stejně jako u kompresních technik známých např. u obrázků nebo videí, i u prostorových dat můžeme použít kompresi ztrátovou, či neztrátovou, podle aplikace. Někdy nám záleží na přesné reprezentaci objektů a nemůžeme si dovolit jakékoli zkreslení (např. u lékařských aplikací), v jiných případech jde především o rychlost a malé odchylky při zobrazení nehrají pro uživatele velkou roli

(např. 3D televize).

Cest jak zmenšit množství přenášených dat je tedy více, ale co zatím nebylo příliš v popředí pozornosti jsou možnosti porovnání takových technik. A právě zavedení a srovnání přístupů, jak vyhodnotit výsledky kompresních, potažmo zjednodušujících algoritmů je náplní této práce. Všechny výpočetní metody, které porovnávají dvě reprezentace téhož objektu nám poskytují jejich tzv. objektivní srovnání. Použitím tedy dostaneme přesný údaj o podobnosti povrchů, ale jak ukážeme, tato objektivní metrika se může velmi lišit od toho, jak výsledek vnímá pozorovatel, tedy od subjektivního pohledu člověka. Budeme se tedy zabývat i touto stránkou věci a zavedeme objektivní a subjektivní srovnání metrik.

1.1 Struktura dokumentu

V kapitole 2 se budeme věnovat technikám pro vyhodnocení podobnosti dvou statických sítí a jejich použití pro dynamický případ a zmíníme i metody, které jsou přímo stavěny na srovnání sítí, které se vyvíjí v čase. V dalším oddílu (kap. 3) se podíváme na určité případy, ve kterých uvedené metody pravděpodobně nebudou spolehlivé vzhledem k subjektivnímu srovnání dynamických sítí pozorovatelem. Čtvrtá kapitola se zaměřuje na implementaci metrik. Konkrétnímu porovnání uvedených metod pro různé příklady objektů z hlediska lidského vnímání se pak bude věnovat kapitola 5. V závěru uvedeme souhrn dosažených výsledků.

1.2 Pojmy a značení

Problém, který budeme řešit tedy spočívá v porovnání podobnosti spojitého povrchu \mathcal{S} a jeho modifikované verze \mathcal{S}' , které se vyvíjí v čase. Uvedené algoritmy pracují s diskretní reprezentací těchto povrchů, tedy trojúhelníkovými sítěmi, které označíme jako \mathcal{M} a \mathcal{M}' . Trojúhelníková síť \mathcal{M} je reprezentována množinou \mathcal{P} bodů $p \in \mathbb{R}^3$ a množinou \mathcal{T} trojúhelníků, které definují její strukturu, tedy $\mathcal{M} = (\mathcal{P}, \mathcal{T})$.

Při kompresi obrazu platí, že každý bod zkomprimované verze přísluší ke stejnému bodu původního obrazu. U 3D trojúhelníkové sítě takové omezení znamená, že se

její modifikací nezmění nijak topologie a tedy každý bod lze namapovat jedna k jedné v originální a komprimované (popř. zjednodušené) síti. Pokud toto platí, pak říkáme, že je zachována *konektivita* sítě.

Pro případ dynamického povrchu, popř. trojúhelníkové sítě, budeme používat označení \mathcal{S}_D , popř. \mathcal{M}_D , přičemž dynamickou sítí budeme rozumět sekvenci trojúhelníkových sítí s konstantní konektivitou v čase.

Zavedeme si také zkratky pro označení jednotlivých metrik, které budeme dále používat, především v tabulkách a grafech:

MSE	Mean Squared Error
HD	průměrná Hausdorffova vzdálenost
HD RMS	Hausdorffova vzdálenost Root Mean Squared Error
KG	Karniho a Gotsmanova metrika
VCHV	metoda vážených chybových vektorů s topologickou sousedností
VCHVs	metoda vážených chybových vektorů s prostorovou sousedností
TD	metoda odchylek trojúhelníka (trojúhelníková diference)
4D HD	průměrná 4D Hausdorffova vzdálenost
4D HDp	největší 4D Hausdorffova vzdálenost
RibbonA	Ribbon metrika - průměrná odchylka
RibbonP	Ribbon metrika - maxiální odchylka.

Kapitola 2

Metriky pro porovnání trojúhelníkových sítí

Jak již bylo zmíněno, existují techniky, které se zabývají měřením odchylky statických povrchů [Aspe02, Cign96]. Tyto metody tedy definují funkci $d(\mathcal{S}, \mathcal{S}')$, která je metrikou podobnosti těchto povrchů. Zmíníme tři takové metody, kterými jsou *MSE*, *Hausdorffova vzdálenost* a metoda uvedená v práci Karniho a Gotsmana ([Karn04]) (dále ji budeme označovat jako *metodu KG*). Do stejné skupiny se řadí také nově uváděná metoda *trojúhelníkových diferencí (TD)*. Jejich aplikací získáme porovnání povrchů v určitém okamžiku a pro jejich srovnání v čase, tedy $d(\mathcal{S}_D, \mathcal{S}'_D)$, použijeme průměr těchto hodnot v každém snímku celé animace.

Dále uvedeme dvě metody (*4D Hausdorffova vzdálenost* [VaSk06] a tzv. *ribbon metriku*, která je součástí standardu MPEG), které srovnávají přímo dynamické povrchy.

Všechny zmíněné metody mají své přednosti, ale také nedostatky. A s řešením některých těchto problémů se snaží vypořádat nová metoda, kterou v této práci zavedeme. Je založena na výpočtu vektorů odchylek jednotlivých bodů dynamických sítí a pro její označení tedy budeme používat termín *vážené chybové vektory - VCHV*. Tato metrika se řadí do druhé uvedené skupiny metrik, tedy bere při výpočtu v úvahu časový průběh a vývoj sítě.

2.1 Mean Squared Error

Ve statistických aplikacích, kdy odhadujeme hodnotu nějaké náhodné veličiny, se pro zjištění očekávané odchylky odhadu používá metrika známá jako *mean squared error* (*MSE*). Pokud označíme skutečnou hodnotu veličiny jako x a i -tou realizaci odhadu této proměnné \hat{x}_i , pak hodnota odchylky *MSE* se spočte

$$MSE(\hat{x}) = E [(\hat{x}_i - x)^2] \quad \text{pro } i = 1..n \quad (2.1)$$

kde E značí střední hodnotu a n je počet realizací náhodné veličiny.

Dále se zavádí metrika ve statistice používaná pod názvem směrodatná odchylka (*root mean squared error*, dále jen *RMSE*)¹. Ta je definována jako odmocnina z *MSE*, tedy

$$RMSE(\hat{x}) = \sqrt{E [(\hat{x}_i - x)^2]}. \quad (2.2)$$

Stejně jako odhadovaná veličina, tak i *MSE* (popř. *RMSE*) je náhodnou proměnnou a jako taková musí být její hodnota odhadnuta, většinou aritmetickým průměrem jednotlivých odchylek

$$MSE = \frac{1}{n} \sum_i (\hat{x}_i - x)^2. \quad (2.3)$$

Technika *MSE* je nejčastěji používanou metrikou v oblasti zpracování obrazu, např. pro vyhodnocení kvality rekonstrukce poškozeného obrazu, kdy se počítá průměrná hodnota odchylek v každém pixelu obrazu. Většinou se ale nepoužívá v té formě, jak jej uvádí (2.3), ale používá se zápis známý pod zkratkou *PSNR* (*Peak Signal-to-Noise Ratio*)

$$PSNR = 10 \cdot \log \left(\frac{max_I^2}{MSE} \right) = 20 \cdot \log \left(\frac{max_I}{RMSE} \right), \quad (2.4)$$

kde max_I je maximální intenzita pixelu v obrazu (u šedotónových obrázků je to 255).

My stejný postup jako u bodů obrazu použijeme pro naše povrchy \mathcal{S} a \mathcal{S}' a budeme počítat průměr odchylek jednotlivých bodů p' oproti původním bodům p . Výpočet

¹Někdy je možné se setkat i s označením *root mean squared deviation* (*RMSD*)

hodnoty MSE tedy bude

$$MSE = \frac{1}{n} \sum_{p \in \mathcal{S}} (p - p')^2 \quad (2.5)$$

a stejně tak bude podobný vzorec pro PSNR

$$PSNR = 10 \cdot \log \left(\frac{\Delta \mathcal{S}^2}{MSE} \right), \quad (2.6)$$

kde $\Delta \mathcal{S}$ je diagonála modelu.

Uvedená metoda je velice jednoduchá a výpočetně rychlá. Bohužel tato jednoduchost má za následek mnoho nedostatků. Jelikož se při výpočtu odchylky nijak nezohlednuje její směr, může se stát, že odchylka v povrchu, kterou lidský pozorovatel téměř nepostřehne vyjde v porovnání lépe než jiná nerovnost, která bude pro člověka velmi rušivá. Ještě větším problémem této metriky ale je, že ji nelze použít na porovnání povrchů, pokud mezi nimi není zachována konektivita.

2.2 Hausdorffova vzdálenost

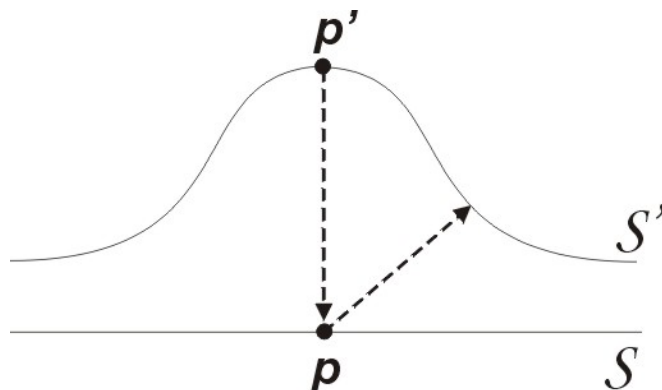
Omezení, které představuje zachování konektivity, jak je tomu u metody MSE je u 3-rozměrných modelů velmi zásadní a v praxi je takový postup prakticky nepoužitelný, zvláště pro algoritmy zjednodušení sítí, které předpokládají změnu topologie a tedy konektivity. Nelze tedy jednoduše měřit odchylku bodu jako vzdálenost jeho původní a modifikované verze. Abychom mohli aplikovat koncept metriky MSE, potřebujeme tedy k libovolnému bodu $p \in \mathcal{S}$, najít bod $p' \in \mathcal{S}'$, který s ním asociujeme. K tomu využijeme princip *Hausdorffovy vzdálenosti* [Aspe02, Cign96], která definuje vzdálenost mezi dvěma povrchy.

Nejprve si nadefinujeme vzdálenost $d(p, \mathcal{S}')$ mezi bodem p , který leží na povrchu \mathcal{S} a modifikovaným povrchem \mathcal{S}' jako

$$d(p, \mathcal{S}') = \min_{p' \in \mathcal{S}'} \|p - p'\|, \quad (2.7)$$

přičemž $\|\bullet\|$ označuje Euclidovskou normu. Hausdorffova vzdálenost mezi povrchy \mathcal{S} a \mathcal{S}' pak bude

$$d(\mathcal{S}, \mathcal{S}') = \max_{p \in \mathcal{S}} d(p, \mathcal{S}') \quad (2.8)$$



Obrázek 2.1: Ukázka asymetričnosti Hausdorffovy vzdálenosti. Vzdálenost $d(p, \mathcal{S}')$ je zjevně mnohem menší, než $d(p', \mathcal{S})$.

Je nutno zmínit, že tato vzdálenost není symetrická, tedy neplatí $d(\mathcal{S}, \mathcal{S}') = d(\mathcal{S}', \mathcal{S})$. To lze ilustrovat na příkladu dvou sítí na obr.2.1. Jak je vidět, tak v případě, že bychom uvažovali pouze jednosměrnou vzdálenost obou povrchů, mohli bychom dostat velmi zavádějící hodnoty. Proto se zavádí tzv. *symetrická Hausdorffova vzdálenost* $d_S(\mathcal{S}, \mathcal{S}')$, která je definována jako

$$d_S(\mathcal{S}, \mathcal{S}') = \max(d(\mathcal{S}, \mathcal{S}'), d(\mathcal{S}', \mathcal{S})). \quad (2.9)$$

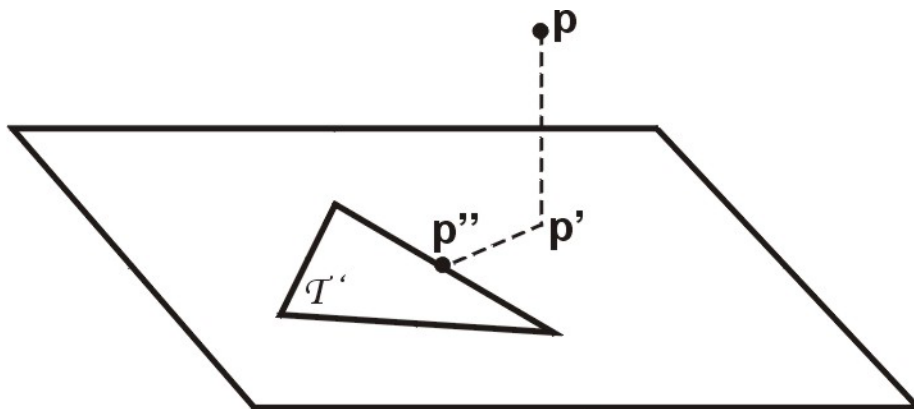
Funkce $d(\mathcal{S}, \mathcal{S}')$ bývá označována jako *dopředná* a funkce $d(\mathcal{S}', \mathcal{S})$ jako *zpětná vzdálenost*.

Pouze hodnota největší vzdálenosti dvou bodů p a p' nám ale o podobnosti obou povrchů \mathcal{S} a \mathcal{S}' příliš neřekne. Jak bylo zmíněno výše, použijeme tedy princip MSE, jen místo odchylky $d(p, p')$ dosadíme vzdálenost bodu p od povrchu \mathcal{S}' , tedy $d(p, \mathcal{S}')$, jak je nadefinována v (2.7). Střední odchylku pro dva spojitě povrchy \mathcal{S} a \mathcal{S}' pak můžeme zapsat následovně

$$d_m(\mathcal{S}, \mathcal{S}') = \frac{1}{|\mathcal{S}|} \int_{p \in \mathcal{S}} d(p, \mathcal{S}')^2 d\mathcal{S}, \quad (2.10)$$

kde $|\mathcal{S}|$ označuje celkovou plochu povrchu \mathcal{S} . Stejně jako u MSE potom můžeme vypočítat i směrodatnou odchylku RMSE jako

$$d_{rmse}(\mathcal{S}, \mathcal{S}') = \sqrt{\frac{1}{|\mathcal{S}|} \int_{p \in \mathcal{S}} d(p, \mathcal{S}')^2 d\mathcal{S}}. \quad (2.11)$$



Obrázek 2.2: Vzdálenost bodu p a trojúhelníku \mathcal{T}' jako vzdálenost p a nejbližšího bodu $p'' \in \mathcal{T}'$.

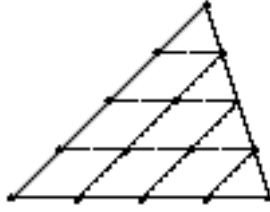
Stejně jako u (2.9) můžeme stejným způsobem dodefinovat symetrické varianty těchto odchylek.

Z hlediska implementace je pro nás ale důležitější případ diskrétních povrchů, tedy budeme chtít zjistit hodnoty rovnic (2.10), (2.11) pro trojúhelníkové sítě \mathcal{M} , \mathcal{M}' . Pak problém nalezení minimální vzdálenosti (2.7) bude zřejmě problémem nalezení minimální vzdálenosti bodu p ležícím na síti \mathcal{M} a trojúhelníků $T' \in \mathcal{T}'$.²

Vzdálenost bodu k trojúhelníku lze spočítat analyticky. V případě, že pravoúhlý průmět bodu p do plochy trojúhelníku \mathcal{T}' leží uvnitř tohoto trojúhelníku, pak jde jednoduše o vzdálenost bodu a plochy. V opačném případě se bude vzdálenost $d(p, \mathcal{T}')$ rovnat vzdálenosti dvou bodů p a p'' , kde p'' je nejbližší bod trojúhelníku \mathcal{T}' k bodu p a tedy leží na jedné z hran trojúhelníku (viz. Obr.2.2).

Teoreticky by bylo možné spočítat vzdálenost $d(p, \mathcal{S}')$ pro libovolný bod p , ale pro implementaci je třeba síť \mathcal{M} nějak navzorkovat. Aspert a kol. ([Aspe02]) zavedli pravidelné vzorkování tak, jak naznačuje obr.2.3. Každý trojúhelník z množiny \mathcal{T} je vzorkován tak, že každou stranu trojúhelníka rozdělíme n vzorky a spojením vzniklých uzlů na dvou stranách s uzly na třetí straně vznikne pravidelná mřížka, jejíž uzly vezmeme jako vzorky, ve kterých se budou počítat vzdálenosti k síti \mathcal{M}' .

²Je nutno si uvědomit, že bod p nemusí být prvkem \mathcal{P}



Obrázek 2.3: Ukázka mřížky, která vzorkuje trojúhelník pro počet vzorků na straně trojúhelníka $n = 5$. (Převzato z [Aspe02])

Nyní zbývá dopočíst vzdálenosti (2.10), (2.11). Označíme si čtyři sousední vzorky jako $x_{i,j}$, $x_{i+1,j}$, $x_{i,j+1}$, $x_{i+1,j+1}$ a $e_{i,j}$, $e_{i+1,j}$, $e_{i,j+1}$, $e_{i+1,j+1}$ budou hodnoty odchylek asociovaných s těmito body (viz. Obr.2.4). Pak hodnoty integrálů získáme součtem podílů všech lichoběžníků a hraničních trojúhelníků, které vznikly navzorkováním trojúhelníku. Lichoběžník, který je tvořen těmito čtyřmi vzorky lze dále rozdělit na dva trojúhelníky, jak je vidět na obr.2.4. Pokud si vezmeme jeden tento navzorkovaný trojúhelník $\mathcal{T}_{ij} = (x_{i,j}, x_{i+1,j}, x_{i,j+1})$ a jejich odchylky, pak nejjednodušší způsob výpočtu integrálu přes tento trojúhelník je lineární interpolace hodnot e , tedy

$$\int_{\mathcal{T}} e = |\mathcal{T}_{ij}| \cdot \frac{e_{i,j} + e_{i+1,j} + e_{i,j+1}}{3} \quad (2.12)$$

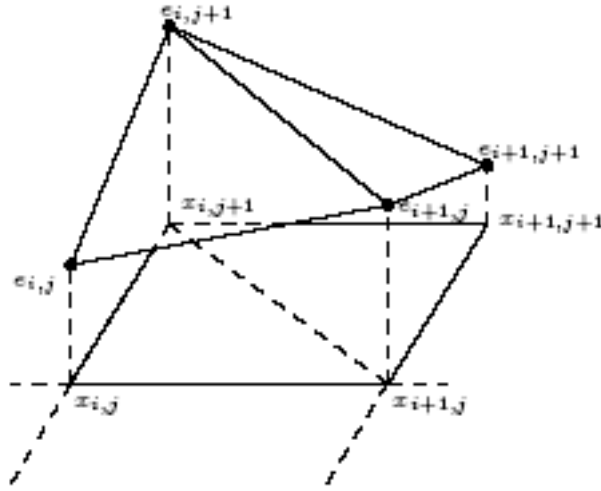
Jestliže budeme chtít spočítat integrál kvadratické odchylky e^2 , pak jeho řešení povede na kvadratickou interpolaci hodnot odchylek a výsledná hodnota bude

$$\int_{\mathcal{T}} e^2 = |\mathcal{T}_{ij}| \cdot \frac{e_{i,j}(e_{i,j} + e_{i+1,j} + e_{i,j+1}) + e_{i+1,j}(e_{i+1,j} + e_{i,j+1}) + e_{i,j+1}^2}{6} \quad (2.13)$$

Použitím Hausdorffovy vzdálenosti jsme oproti MSE (příp. PSNR) metrice odstranili problém ve formě potřeby konstantní konektivity mezi porovnávanými sítěmi. Nicméně další nedostatek zůstává a tím je nezohlednění směru odchylky.

2.3 KG metrika

Karni a Gotsman ve své práci o kompresi animačních sekvencí 3D sítí [Karn04] uvádí metriku, s jejíž pomocí vyhodnocují dosažené výsledky použitých kompresních



Obrázek 2.4: Výpočet hodnoty integrálu odchylky e . (Převzato z [Aspe02])

metod. Vezmeme-li opět dvě množiny bodů \mathcal{P} , \mathcal{P}' , které tvoří dynamické sítě \mathcal{M}_D , \mathcal{M}'_D , pak nadefinujeme tři matice. Pokud M vezmeme jako počet bodů sítě a N počet snímků animace, pak tyto matice budou všechny o rozměrech $3M \times N$. Pro původní animaci poté sestavíme matici A , jejíž sloupce jsou naplněny souřadnicemi všech bodů vždy v určitém snímku animace. Matice A' je pak vytvořena naprosto stejným způsobem, jen tentokrát pro modifikovanou verzi dynamické sítě.

$$A = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ y_{11} & y_{12} & \cdots & y_{1n} \\ z_{11} & z_{12} & \cdots & z_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \\ y_{m1} & y_{m2} & \cdots & y_{mn} \\ z_{m1} & z_{m2} & \cdots & z_{mn} \end{bmatrix} \quad (2.14)$$

Konečně, matice, kterou označíme jako $E(A)$ vznikne tak, že v každém snímku spočítáme průměrnou hodnotu pro jednotlivé souřadnice x , y , z bodů a ty pak M -násobným

opakováním vytvoří sloupec matice.

$$E(A) = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_n \\ \bar{y}_1 & \bar{y}_2 & \cdots & \bar{y}_n \\ \bar{z}_1 & \bar{z}_2 & \cdots & \bar{z}_n \\ \vdots & \vdots & \ddots & \vdots \\ \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_n \\ \bar{y}_1 & \bar{y}_2 & \cdots & \bar{y}_n \\ \bar{z}_1 & \bar{z}_2 & \cdots & \bar{z}_n \end{bmatrix} \quad (2.15)$$

Míra distorze lze potom vyjádřit následujícím výrazem

$$e = 100 \frac{\|A - A'\|}{\|A - E(A)\|} \quad (2.16)$$

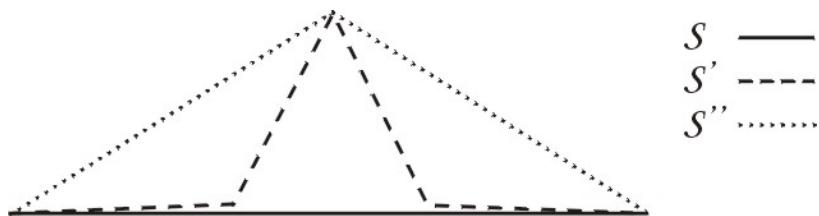
kde $\|\bullet\|$ je Eukleidovskou normou. Násobek stem používají autoři pro vyjádření deformace povrchu v procentech.

Pokud se na vzorec výpočtu celkové odchylky podíváme, ihned odhalíme velké slabiny tohoto přístupu. Především vzorec nijak nezohledňuje délku zpracovávané animace, takže pokud budeme mít dvě dynamické sítě stejného modelu s konstatní distorzí, jen jedna bude mít poloviční délku trvání, dostaneme s použitím této metriky zcela odlišné výsledky, i když bychom spíše očekávali, že hodnoty budou vzhledem ke stejné míře distorze podobné. Více o invariantnosti metod vůči různým parametrům dynamické sítě uvedeme v kap. 3.3.

2.4 Vážené chybové vektory

Touto novou metodou se snažíme odstranit některé problémy, který předchozí metriky neřeší. Ukázka takové situace je naznačena na obr.2.5, kde je zobrazen jak původní povrch \mathcal{S} , tak jeho dvě možné aproximace \mathcal{S}' a \mathcal{S}'' . V tomto případě tyto metriky selžou a oproti subjektivnímu dojmu pozorovatele prohlásí povrch \mathcal{S}' za lepší aproximací \mathcal{S} , než je \mathcal{S}'' . Neberou totiž v úvahu povahu odchylky a tedy, že jednorázový skok je mnohem rušivější, než celkový posun povrchu.

Myšlenka, jak se s tímto vypořádat, spočívá v zahrnutí odchylek sousedních bodů do výpočtu metriky v daném bodě. Podíváme se tedy po okolí a porovnáme, jak moc



Obrázek 2.5: Nedostatek metrik PSNR a Hausdorffova vzdálenost. Porovnáním povrchů \mathcal{S} , \mathcal{S}' a $\mathcal{S}, \mathcal{S}''$ s použitím těchto metrik zjistíme, že povrch \mathcal{S}' je lepší aproximací \mathcal{S} . Pro pozorovatele však bude přijatelnější \mathcal{S}'' , jelikož neobsahuje náhlý skok.

se odchylky liší. Pokud jsou v celém okolí podobné, pak se dá čekat, že ve výsledku nebudou příliš rušivé, protože nedochází k nějakému kolísání či skokům.

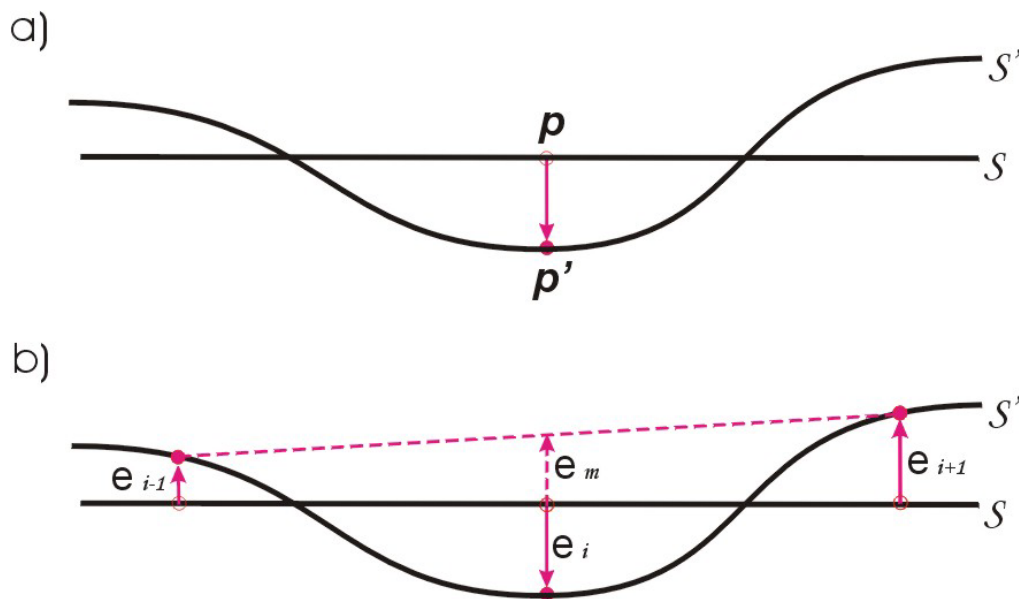
Nejprve si zavedeme chybový vektor $e = (p' - p)$, který určí směr a velikost odchylky každého bodu $p \in \mathcal{M}$ (viz.obr.2.6 a). Pro $p = (x, y, z)$ tedy $e = (\Delta x, \Delta y, \Delta z)$.

Jakmile máme tyto vektory spočtené ve všech bodech sítě, vyhodnotíme metriku v každém bodu p následovně (viz.obr.2.6 b):

1. vezmeme všechny topologicky sousední body k bodu p a spočteme průměr chybových vektorů s nimi asociovanými $e_m = \frac{1}{n} \sum e_i$, přičemž n je počet sousedů. Jelikož v síti nejsou body rozmístěné pravidelně, lze při výpočtu aplikovat navíc vážený průměr na základě vzdálenosti jednotlivých sousedů od p ;
2. výsledná odchylka, kterou přiřadíme k bodu p bude $d(p, p') = |e - e_m|$.

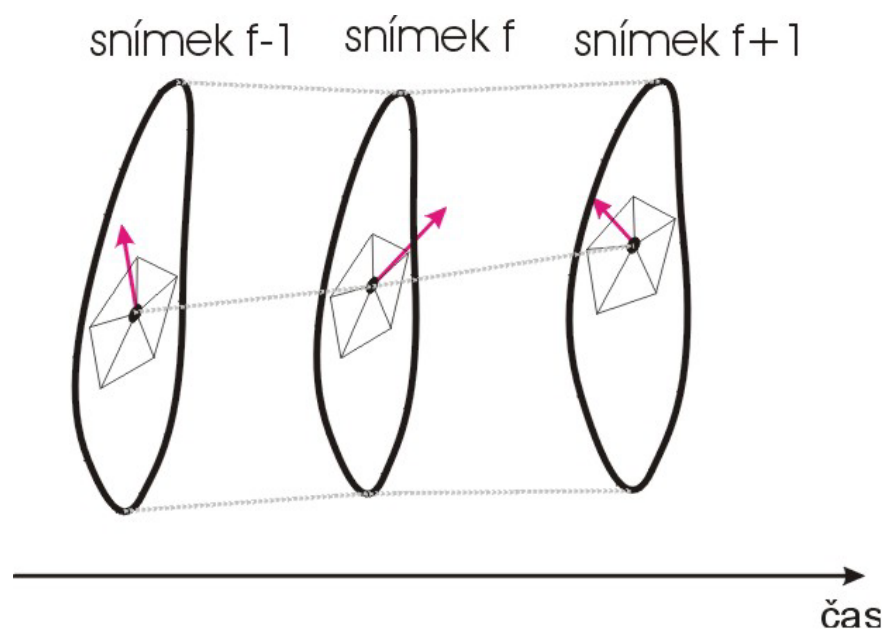
Výslednou hodnotu pro celou síť získáme opět průměrem odchylek ze všech jejích bodů.

Aplikací této metody vyřešíme zmíněný problém se směrem odchylek, ale vrátili jsme se zpět k nutnosti zachování konektivity, což představuje velké omezení. Otázkou je, zda bychom nemohli využít stejný postup, jako u PSNR metriky. Když počítáme Hausdorffovu vzdálenost, tak vlastně hledáme k určitému bodu $p \in \mathcal{M}$ ten nejbližší bod $p' \in \mathcal{M}'$. Místo toho, abychom porovnávali vektory posunutí původního bodu tedy vezmeme nejbližší bod p' a chybový vektor budeme počítat vůči němu.

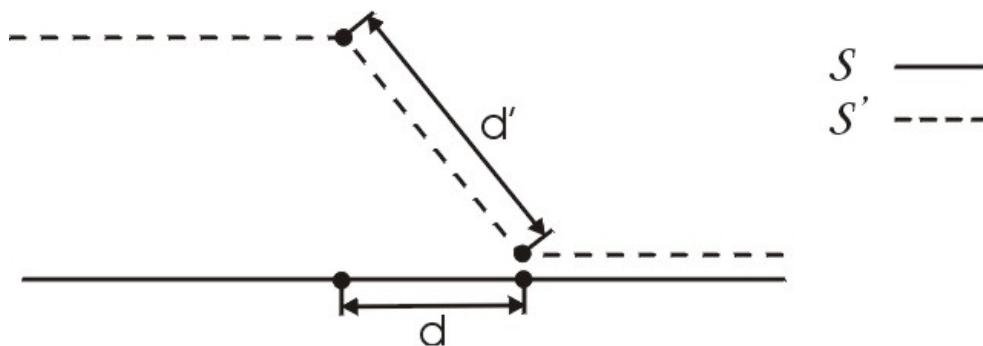


Obrázek 2.6: a) Chybový vektor vzniklý posunem bodu $p \in \mathcal{S}$; b) průměrem sousedních chybových vektorů dostaneme e_m a výslednou odchylku odečtením tohoto vektoru od $(p' - p)$.

Popsaným způsobem tedy můžeme získat odchylku každého bodu v určitém snímku animace. Stejně jako u předchozích metod bychom nyní mohli jednoduše tyto hodnoty zprůměrovat přes počet bodů a snímků a dostali tak celkovou hodnotu metriky pro dynamickou síť. Tím bychom ale napodobovali chování metod PSNR a Hausdorffovy vzdálenosti a případné artefakty vzniklé v čase by zůstávaly neodhaleny. Elegantním řešením by mohla být aplikace principu chybových vektorů i na časovou složku. Poté, co jsme spočetli v každém snímku odchylky jednotlivých bodů povrchu, můžeme pro snímek vzít odchylky bodu v okolních snímcích a porovnat jejich průměr s diferencí našeho bodu (obr. 2.7). Jako vstupní parametr metriky by se udávala hloubka, do jaké až prohledávat časové sousedy aktuálního snímku. Průměrem získáme hodnotu odchylky pro bod přes celou dynamickou síť a celkovou odchylku spočítáme opět jako průměr přes všechny body.



Obrázek 2.7: Chybový vektor vzniklý posunem bodu $p \in \mathcal{S}$ v aktuálním snímku a jeho bezprostředních časových sousedech. Výsledný chybový vektor bodu p pro snímek f vznikne jeho porovnáním s průměrem chybových vektorů sousedů (snímky $f - 1$ a $f + 1$).



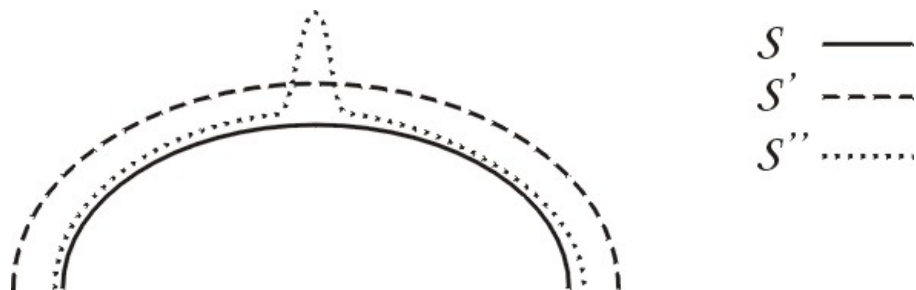
Obrázek 2.8: Metoda trojúhelníkové difference ve 2D. Porovnáváme délky úseků d a d' , takže odchylka jednoho úseku je rovna $|d - d'|$.

2.5 Trojúhelníková difference

Všechny dosud zmíněné metriky, i když každá odlišně, vychází ze srovnání odchylek jednotlivých bodů původní a upravené trojúhelníkové sítě. Otázka je, zda by se nedalo při výpočtu využít místo bodů také jiných elementů sítě. O to se snaží metoda trojúhelníkových diferencí, kterou vyvinul autor v rámci této práce. Jako nejmenší jednotka se v tomto přístupu nebere bod sítě, nýbrž jedna ploška, tedy trojúhelník.

Vezměme si nejprve dvojrozměrný případ, kdy porovnáváme \mathcal{S} a \mathcal{S}' (obr. 2.8). Body sítě vytvářejí souvislou posloupnost lineárních úseků. My budeme chtít vzít délku každého takového úseku a porovnat ji u původní a upravené sítě. Absolutní hodnotu rozdílu těchto délek označíme jako odchylku daného úseku. Průměrem přes všechny úseky získáme celkovou odchylku. Přejít do třech dimenzí pak bude znamenat to, že místo délek dvou úseček budeme porovnávat obsahy trojúhelníkových plošek modelu.

Ačkoliv se tato metoda úspěšně vypořádá s většinou artefaktů, které ostatním metodám dělají problémy (kap. 3), i ona má svá slabá místa. Vezměme si např. model kulového tvaru, který zdeformujeme dvěma různými způsoby. V prvním případě pouze zvětšíme poloměr koule, v druhém pak poloměr zachováme, ale na povrchu se vyskytne náhlý skok (viz. obr. 2.9). V takové situaci si lidský pozorovatel pravděpodobně vůbec nevšimne změny velikosti, zatímco výstupek na jinak hladkém povrchu



Obrázek 2.9: Původní povrch \mathcal{S} deformujeme dvěma způsoby. Zatímco pro pozorovatele bude přijatelnější \mathcal{S}' , metoda TD prohlásí za lepší variantu \mathcal{S}'' .

bude působit velmi rušivě. Měřením změny obsahů trojúhelníkových plošek však dojdeme k opačnému závěru a za přijatelnější prohlásíme druhý případ. Pravdou ale je, že při kompresi se s takovým defektem pravděpodobně setkávat nebudeme.

Mnohem zásadnějším omezením metriky je fakt, že vyžaduje zachování konektivity a není přímočará cesta, jak tento nedostatek obejít. Pro algoritmy zjednodušující dynamické sítě tedy není vhodná. Metoda také nebere přímo v úvahu časově závislé chyby. Také v tomto případě bychom mohli využít podobného principu, jako u metody chybových vektorů, kdy se berou v úvahu odchylky v časových sousedech aktuálně zpracovávaného snímku. Takové rozšíření může být předmětem další práce.

2.6 4D Hausdorffova vzdálenost

Tato metoda, která byla prezentována ve [Vasa06] se na rozdíl od všech předešlých snaží porovnávat přímo dynamické trojúhelníkové sítě, namísto statických. Vychází z myšlenky Hausdorffovy vzdálenosti, ale přidává další rozměr a tím je čas. Budeme tedy hledat Hausdorffovu vzdálenost mezi sítěmi ve 4D.

Při takovém přístupu se zjevně musíme potýkat s velkým problémem a to, že nelze použít stejné jednotky pro všechny dimenze. Předpokladem je, že stejná odchylka v jakémkoliv rozměru by měla být v naší reprezentaci 4D sítě rovna stejné vzdálenosti. Potřebujeme tedy najít nějaký vztah mezi časovou a prostorovou složkou. Navíc, zatímco čas je velmi přesně vyjádřen, tak u prostorových jednotek v počítačovém modelování není většinou jasné, o jaké jednotky se vlastně jedná. Potřebujeme

tedy použít prostorovou jednotku relativní vzhledem k modelu a tu zavedeme jako vzdálenost v modelu ku velikosti jeho diagonály. S použitím této definice lze očekávat srovnatelné výsledky u libovolného modelu.

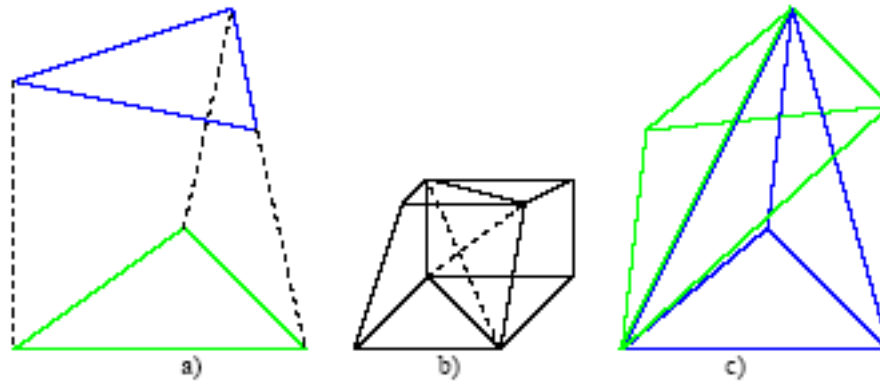
Dále potřebujeme zavést nějakou konstantu, označíme ji α , která vyjádří vztah mezi prostorovou a časovou složkou. Pro zjištění vyhovující hodnoty této konstanty je potřeba provést experimentální měření závislé na vnímání pozorovatele. Můžeme si ale zavést některé předpoklady na základě znalosti lidského vnímání, které nám pomohou alespoň odhadnout hodnotu této konstanty:

1. pro člověka je časový úsek 1/100s téměř nerozeznatelný a zároveň zhruba 10% prostorový posun je na hraně únosnosti. Z toho se dá odhadnout, že α bude menší než $\frac{0.1}{0.01} = 10$;
2. časový úsek jedné vteřiny je z hlediska nenarušení vnímání někde na hranici, zatímco prostorovou odchylku menší jak 0,1% už pro lidské oko nepostřehne. Pokud tyto dvě hodnoty opět poměříme, α by měla být větší než $\frac{0.001}{1} = 0.001$.

Dá se očekávat, že ideální hodnota konstanty α se bude pohybovat někde u středu vymezeného intervalu, tedy okolo 0.1.

Nyní, když máme sjednocené jednotky, musíme zkonstruovat z prostorové reprezentace modelu 4-rozměrnou síť. Toho dosáhneme tak, že z každého trojúhelníku v dvou po sobě jdoucích snímcích vytvoříme časoprostorový hranol (obr.2.10a). Stěny tohoto hranolu ale nejsou rovinné. Rozdělíme tedy každou boční stěnu diagonálou a hranol poté můžeme rozdělit na tři tetrahedrony, jak je naznačeno na obr.2.10c. Směr diagonál je potřeba zachovávat konzistentní vůči sousedním hranolům, abychom získali spojitou síť bez děr. Proto se zavádí následující systém dělení hranolu, který správný směr diagonál zaručí:

1. první tetrahedron se vytvoří ze třech bodů tvořících základnu hranolu a z jednoho bodu z horní podstavy hranolu, který má největší index;
2. druhý tetrahedron vytvoří všechny tři body horní podstavy hranolu a bod základny s nejnižším indexem;



Obrázek 2.10: a) 4D hranol vytvořený z trojúhelníku pohybujícího se v čase; b) dva sousední hranoly v síti a dvojí možné vedení diagonály na jejich boční stěně; c) dělení hranolu na tři tetrahedrony. (Převzato z [Vasa06])

3. poslední tetrahedron vznikne ze dvou bodů základny s nejvyššími indexy a dvou bodů horní stěny s nejnižšími indexy.

Když jsme hledali Hausdorffovu vzdálenost u 3-rozměrných trojúhelníkových sítí, zužoval se problém na výpočet vzdálenosti bod-rovina, příp. bod-bod, když pravoúhlý průmět bodu neležel uvnitř trojúhelníka. Tentokrát budeme muset hledat vzdálenost mezi bodem a tetrahedronem a opět podle polohy pravoúhlého průmětu bodu se výpočet rozdělí na několik možností:

1. průmět bodu leží uvnitř tetrahedronu, pak počítáme vzdálenost bodu k objektu tetrahedronu
2. průmět bodu do roviny stěny tetrahedronu leží uvnitř této stěny, pak počítáme vzdálenost bod-stěna tetrahedronu
3. průmět do roviny stěny tetrahedronu leží na její hraně, pak počítáme vzdálenost bodu k hraně
4. jinak počítáme vzdálenost bodu k bodu tetrahedronu.

Ani u této metody samozřejmě není omezení v nutnosti zachování konektivity. Výhodou pak určitě je zapracování časové složky do výpočtu metriky, takže časové

odchyly jsou vyhodnocovány stejně, jako ty prostorové. Časové nároky na výpočet se dají do jisté míry snížit pokročilejšími technikami preprocessingu a dělením prostoru ([Vasa06]), nicméně pro animace komplexnějších modelů i tak dosahuje výpočetní čas řádu hodin. Největším nedostatkem této metody je ale její paměťová náročnost. Vzhledem k tomu, že konstruuje pro každý snímek animace ke každému trojúhelníku hranol, který vytvoří tři tetrahedrony, je tento přístup použitelný prakticky jen pro malé modely a krátké sekvence, ale i tak může sloužit jako zajímavá metrika pro porovnání.

2.7 Ribbon metrika

Poslední metrika, kterou uvedeme, vychází ze standardu MPEG a je založena na veličině nazývané plošná diference (Area Difference). Tato diference, kterou označíme jako D_A , vyjadřuje průměrný rozdíl mezi dráhou původního bodu $p \in \mathcal{M}$ a dráhou bodu z modifikované sítě, jako obsah plochy, kterou tyto dvě trajektorie vytyčují (viz. obr.2.11).

V závislosti na tom, jestli se trajektorie ve dvou po sobě jdoucích snímcích překříží, takovou plochu můžeme přibližně poskládat z lichoběžníků, které mohou a nemusí být překroucené (obr.2.11). Pro obsah lichoběžníku, pokud použijeme označení z obr.2.12, platí

$$A_{lich} = \frac{(a + b)h}{2} \quad (2.17)$$

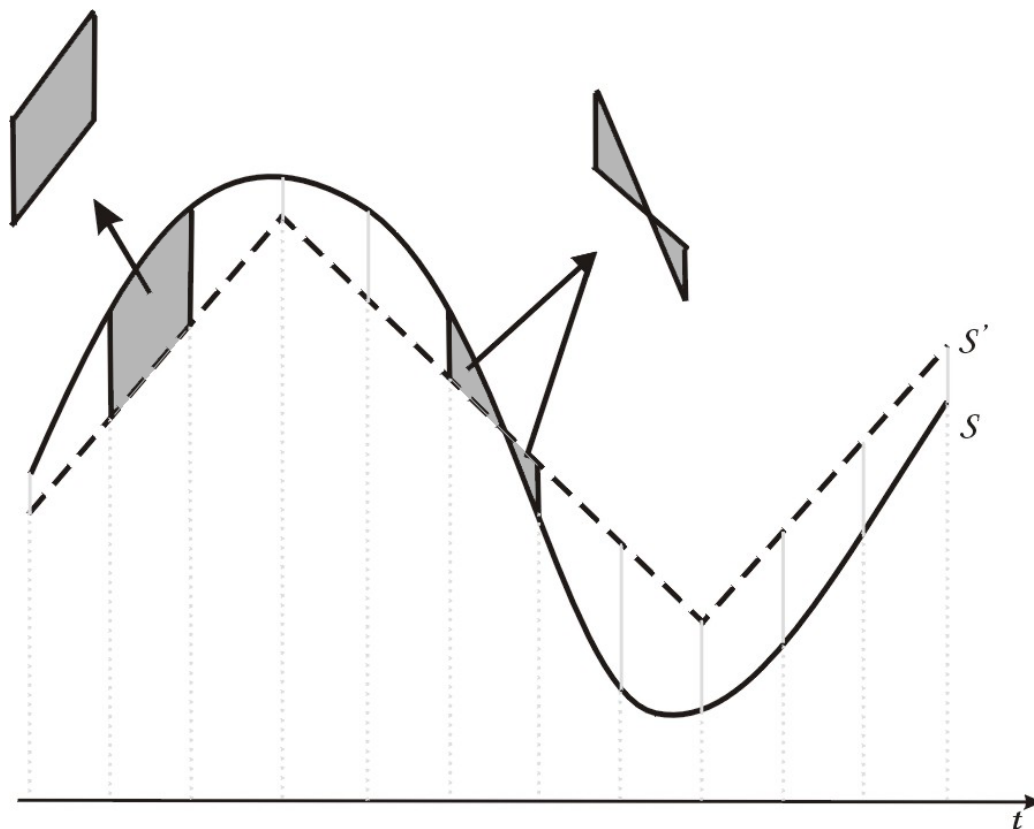
a pro případ překrouceného lichoběžníku

$$A_{twist} = \frac{a^2 + b^2}{2(a + b)}h. \quad (2.18)$$

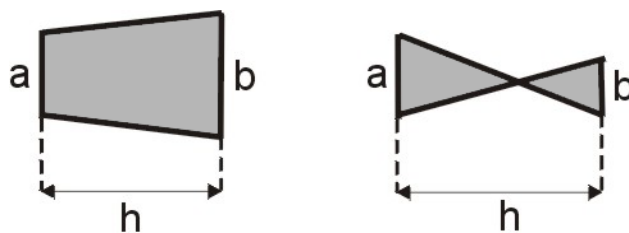
Dále budeme pro oba případy používat společné označení A , bez ohledu na formu lichoběžníku.

Počet bodů v síti označíme M a počet snímků animace N . Celkovou plochu pro jeden bod sítě přes celou animaci tedy získáme součtem obsahů jednotlivých lichoběžníků, takže pro plošnou diferenci j -tého bodu ve směru souřadné osy x bude platit:

$$D_{jX} = \sum_{i=0}^{N-1} A_{i,j} \quad (2.19)$$



Obrázek 2.11: Měření plošné difference mezi trajektoriemi původního a modifikovaného bodu.



Obrázek 2.12: Označení rozměrů normálního a zkrouceného lichoběžníka.

Tuto diferenci samozřejmě můžeme spočítat vzhledem ke každé ze souřadných os, takže dostaneme pro každý bod sítě p_j tři hodnoty D_{jX} , D_{jY} a D_{jZ} .

Stejně jako u některých předchozích metrik, i zde potřebujeme zavést relativní míru, která nebude závislá na rozměrech modelu. Zavedeme tedy dvě konstanty ΔW a Δt . Zatímco první z nich je velikost diagonály modelu přes celou animaci, druhá je vyjádřena jako rozdíl času posledního a prvního snímku, tedy celková doba trvání animace. Výsledná odchylka pro bod sítě se pak spočte jako

$$D_j = \frac{D_{jX} + D_{jY} + D_{jZ}}{3 \cdot \Delta W \cdot \Delta t}. \quad (2.20)$$

Průměrná odchylka D_A animace pak bude rovna průměru všech dílčích odchylek bodů

$$D_A = \frac{1}{M} \cdot \sum_{j=0}^{M-1} D_j. \quad (2.21)$$

Výhodou této metriky je, že porovnává průběh jednotlivých bodů přímo v čase a ne jen průměrem odchylek statických sítí. Bohužel ale nijak nezohledňuje směr odchylky, která v čase nastává a v určitých případech tak nemusí podávat přesné výsledky (viz. kap.3.2).

Jednu důležitou věc jsme však zatím nechali bez povšimnutí. Je jí atribut h ve výpočtu plochy lichoběžníku. Jde vlastně o časovou složku odchylky, bohužel v dostupných materiálech k Ribbon metrice není tato míra nijak definována, i když podávané výsledky jsou na ní zjevně velice závislé. Tato závislost je však při pohledu na vzorce výpočtu odchylky lineární. Pokud tedy použijeme při měření vždy stejnou hodnotu, získáme také konzistentní výsledky.

Kapitola 3

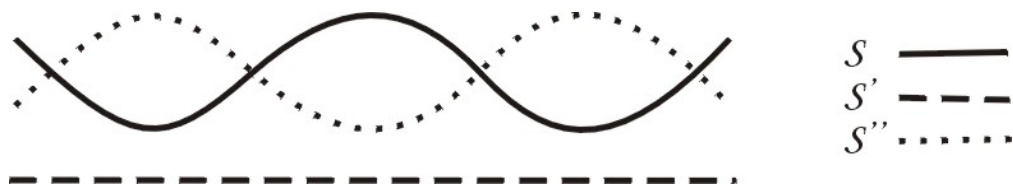
Nedostatky metrik

Zjednodušením či kompresí vzniknou vzhledem k původnímu povrchu určité odchylky. Úlohou metrik zmíněných v předchozí kapitole je objektivně ohodnotit, do jaké míry se původní a modifikovaná verze povrchu liší a přitom by tyto exaktní hodnoty zjištěné výpočtem měly korespondovat se subjektivním pohledem člověka. Otázka tedy je, k jakým deformacím může na povrchu dojít, jak je bude vnímat pozorovatel a jak budou vyhodnoceny metrikami. V této kapitole se zaměříme na druhé zmíněné hledisko, tedy na to, jak zareagují uvedené metriky na různé odchylky povrchu.

Zaveďme si předpoklad o správném chování metrik při porovnávání trojúhelníkových sítí. Vezměme si, že máme tři různé dynamické sítě \mathcal{M}_{D1} , \mathcal{M}_{D2} a \mathcal{M}_{D3} , které budeme porovnávat a funkce $d(\mathcal{M}_{Di}, \mathcal{M}_{Dj})$ definuje metriku odlišnosti sítí. Pak od hodnot spočtených metrikou očekáváme, že bude platit:

1. $d(\mathcal{M}_{D1}, \mathcal{M}_{D2}) = 0 \Leftrightarrow \mathcal{M}_{D1} = \mathcal{M}_{D2}$
2. $d(\mathcal{M}_{D1}, \mathcal{M}_{D2}) = d(\mathcal{M}_{D2}, \mathcal{M}_{D1})$ (3.1)
3. $d(\mathcal{M}_{D1}, \mathcal{M}_{D2}) < d(\mathcal{M}_{D1}, \mathcal{M}_{D3})$ pokud také pro pozorovatele se zdá
být \mathcal{M}_{D2} podobnější \mathcal{M}_{D1} , než \mathcal{M}_{D3}

Z uvedených předpokladů všechny zmíněné metriky splňují první dva body. Pro třetí pravidlo však lze nalézt případy, ve kterých metriky zareagují různě a ne vždy se správným výsledkem. Dále uvedeme několik takových příkladů.



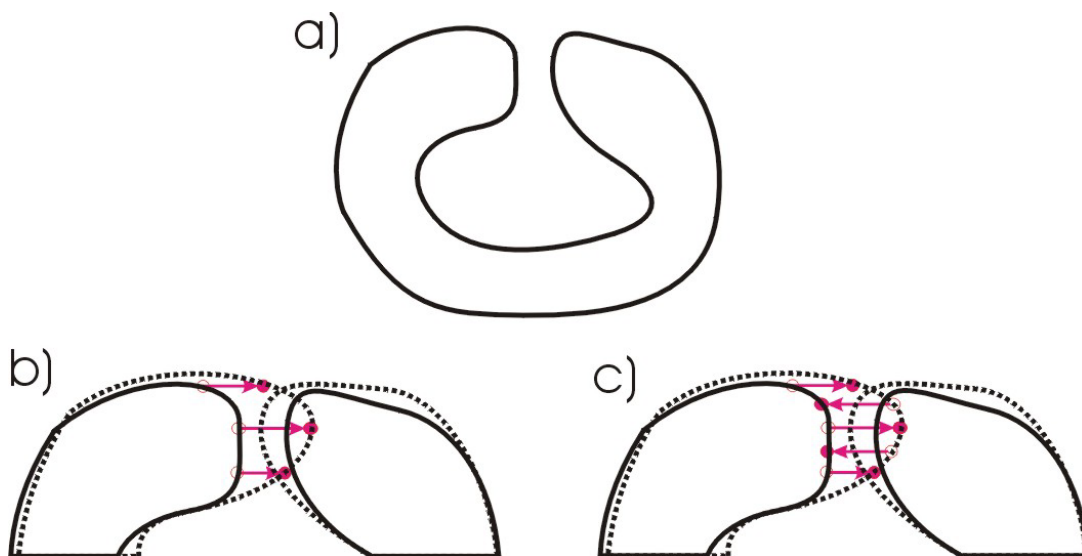
Obrázek 3.1: Problém metody vážených chybových vektorů u zvlněného povrchu. Povrch S' je označen jako lepší aproximace S ačkoliv pro pozorovatele bude mnohem realističtější S'' .

3.1 Odchytky v prostoru

Začneme u příkladu, který už byl zmíněn v kapitole 2.4 a je znázorněn na obr.2.5. Obecně se dá říci, že pokud dojde na původně hladkém povrchu k náhlým skokům či vlnění, pak tato metrika PSNR a metriky založené na Hausdorffově vzdálenosti neodhalí a hodnoty nejsou v takovém případě příliš vypovídající, jelikož povrch, který je jednou z těchto metod prohlášen za dobrou aproximaci se bude pozorovateli jevit jako neakceptovatelný. Stejně tak metoda MPEG nepodá dobré výsledky, jelikož ta řeší pouze časové změny v reprezentaci povrchu, nikoliv prostorové v rámci jednoho snímku. Tento případ ohodnotí správně pouze dvě metody, metoda chybových vektorů a trojúhelníkové diference.

Opačný problém ale vyvstane v případě, že zvlněný je původní povrch a vlivem komprese dojde např. k posunu fáze (obr.3.1). Máme zde původní zvlněný povrch S , další povrch S'' je stejný, jen posunutý a třetí povrch S' vznikl zjednodušením původního a aproximuje ho rovnou plochou. Metoda vážených chybových vektorů je na jakémkoliv kmitání velice citlivá a v tomto případě prohlásí jednoznačně povrch S' za mnohem podobnější původnímu S , než je S'' . Jak je vidět i z obrázku, pro lidské oko bude hodnocení přesně opačné, jelikož S'' zachovává původní charakter povrchu.

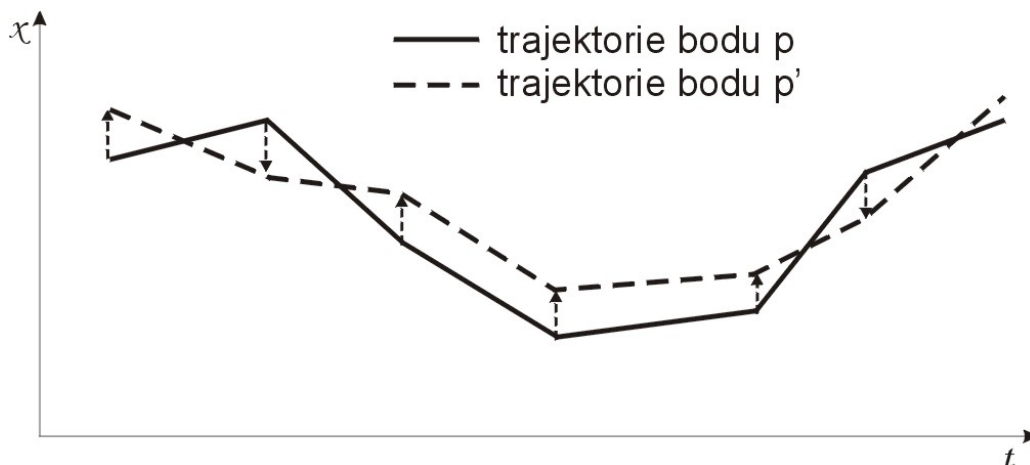
To však není jediný problém u metriky chybových vektorů. Představme si model o tvaru přibližně jako na obr.3.2a. Jestliže tento model upravíme tak, že se oba jeho konce spojí a vytvoří tak jakýsi anuloid, bude to pro pozorovatele nadměrně rušivé a jako takové by to také měla ohodnotit metrika. Podívejme se, jak situaci vyhodnotí výpočet využívající chybové vektory, jak byly popsány. Vzhledem k tomu,



Obrázek 3.2: Problém metody vážených chybových vektorů u ohnutého povrchu. a) původní povrch; b) tečkovaně je naznačen modifikovaný povrch a červené šipky označují chybové vektory, pokud uvažujeme jen topologickou sousednost bodů; c) chybové vektory, pokud uvažujeme prostorovou sousednost vektorů.

že na každém ze spojených konců se všechny body pohybovaly stejným směrem (viz. obr.3.2b), bude hodnota odchylky minimální, což není to, co bychom čekali. Problém je způsoben tím, že se při výpočtu průměrného chybového vektoru berou v úvahu topologičtí sousedé, tedy jen ti, co leží na síti vedle sebe. To by šlo vyřešit tím, že místo topologické sousednosti budeme brát okolí prostorové, tedy nebudeme uvažovat strukturu sítě, ale jen absolutní pozici bodů v prostoru (obr.3.2c). Tak by se do výpočtu vektorů zapojily současně body z obou konců modelu. Tím ale vznikají další problémy s výpočetní složitostí, jelikož tak ztrácíme výhodu datové struktury sítě, která v sobě obsahuje informaci o topologických sousedech. Okolní body tedy musíme hledat v celé množině \mathcal{P} , pravděpodobně s využitím nějaké techniky dělení prostoru.

V tomto případě tedy oproti metodě chybových vektorů vyjdou lépe ostatní techniky. Lze obecně říci, že pro jakýkoliv artefakt vzniklý pouze posunem nějaké části modelu, tzn. všechny body se budou pohybovat stejným směrem, nebude metrika založená na chybových vektorech spolehlivá.



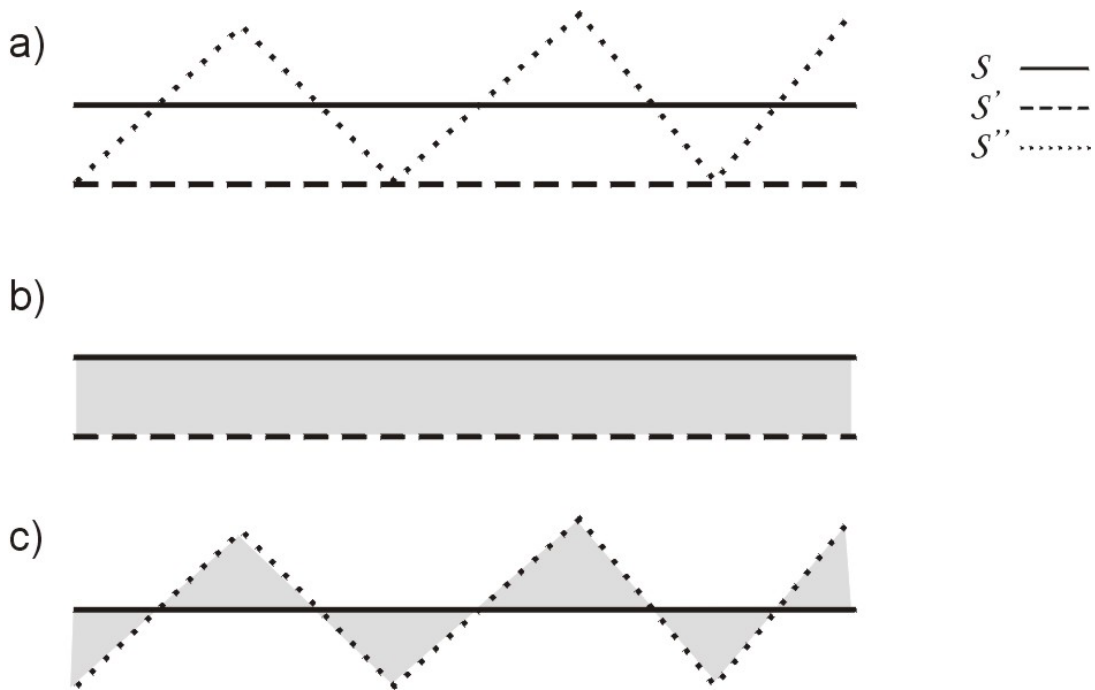
Obrázek 3.3: Trajektorie dvou bodů, které leží na dvou površích, které chceme porovnat. Horizontální osa značí časovou a vertikální osa prostorovou složku polohy bodů.

3.2 Odchyly v čase

Zatím jsme řešili pouze vliv prostorových odchylek v rámci statických sítí. Co se ale stane, pokud se odchylka bodu bude měnit v čase? Jde tedy o to, jak se vyvíjí trajektorie referenčního bodu p a jemu příslušného bodu p' v čase. Pokud se podíváme na obr.3.3, který zobrazuje trajektorie dvou takových bodů, tak zjistíme, že jde vlastně o podobný problém, který jsme řešili u prostorových odchylek. Otázka je, jak se s tím vypořádají metriky, které jsme zavedli.

Metriky založené na MSE, příp. Hausdorffově vzdálenosti, počítají celkovou odchylku dynamické sítě stejným způsobem, jako u statické sítě, tedy z absolutních odchylek bere průměr hodnot. Nemůže tak zachytit, stejně jako ve statickém případě (viz. obr.2.5), některé nežádoucí chování povrchu v čase. To se snaží řešit metoda vážených chybových vektorů aplikací stejného principu výpočtu chyby jak na prostorovou, tak na časovou složku.

Na podobném principu je vlastně založena metoda Ribbon, která počítá odchylky bodů v čase. Neuvažuje ovšem jejich směr v prostoru, takže nezohlední některé artefakty, které mohou být pro pozorovatele velmi rušivé, jako je např. nežádoucí kmitání bodu v čase. To lze ilustrovat situací na obr.3.4. V tomto případě metoda

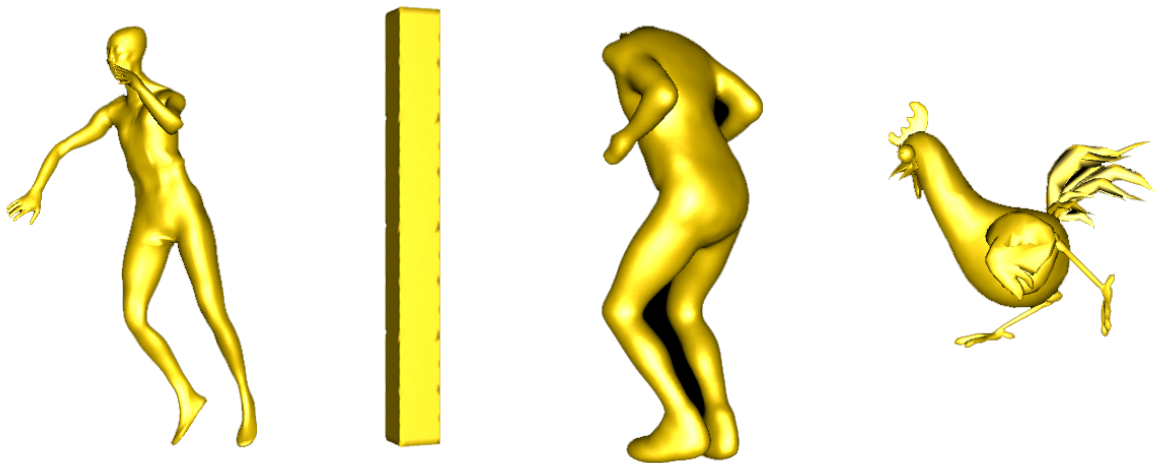


Obrázek 3.4: a) Průběh trajektorie bodu povrchu \mathcal{S} v čase a jeho dvou aproximací \mathcal{S}' , \mathcal{S}'' . b) Plošná diference pro trajektorie bodů povrchu \mathcal{S} a \mathcal{S}' . c) Plošná diference pro trajektorie bodů povrchu \mathcal{S} a \mathcal{S}'' . Aproximace \mathcal{S}'' vyjde z metriky MPEG lépe, ale pro lidské oko bude velmi rušivé kmitání bodu a jako vhodnější bude brát \mathcal{S}' .

Ribbon prohlásí dráhu bodu $p'' \in \mathcal{S}''$ za podobnější původní dráze bodu p , než u bodu $p' \in \mathcal{S}'$. Pozorovatel ale průběhy vyhodnotí zcela naopak, jelikož kmitání bodu v průběhu animace je neakceptovatelné.

3.3 Invariance metrik

Představme si, že pracujeme s modelem objektu, např. hranolu, který se pohybuje ve směru jedné ze souřadných os. Jestliže zachováme stále stejný model hranolu a budeme měnit pouze parametry animace jako celku (jako jsou délka animace či měřítko), pak bychom od hodnot poskytnutých metrikami očekávali, že budou na těchto globálních změnách nezávislé (invariantní). U uvedených metod takové chování však zdaleka není samozřejmostí.

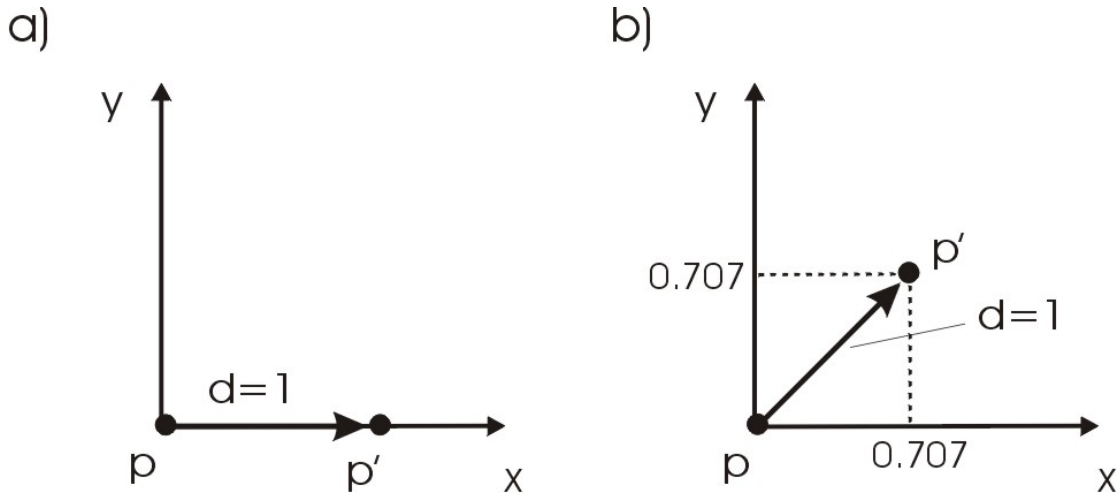


Obrázek 3.5: Ukázky datových sad dynamických sítí použitých v této práci. Zleva *dance*, *hranol*, *human* a *chicken*.

K testování jsme v průběhu práce používali především čtyři datové sady různých délek i počtem bodů. Dále v textu se na ně budeme odkazovat pod názvy *dance*, *hranol*, *human* ([Anua04, Sand03]) a *chicken*. Ukázka těchto dynamických sítí je uvedena na obr. 3.5.

3.3.1 Rotace

Vezměme si jeden bod p , který náleží našemu původnímu modelu hranolu a k němu příslušný bod p' ze stejného modelu, který byl však narušen distorzí. Předpokládejme, že se bod p distorzí posunul do p' o délku $v = 1$ ve směru souřadné osy x (3.6 a). Nyní animaci pozměníme tak, že model pootočíme o $\frac{\pi}{4}$ kolem osy z . Důsledkem toho bude, že se již model nebude pohybovat směrem osy x , ale napříč sektorem vymezeným kladnými poloosami x a y (3.6 b). Nyní budeme chtít zjistit hodnotu odchylky v obou případech. Většina metod uvedených v kap. 2 s tímto nebude mít problém a správně určí pro původní i otočený model odchylku bodu $d = 1$, jelikož tuto odchylku počítá jako vzdálenost dvou bodů v prostoru. Jinak to však bude u metod Ribbon a KG. Tyto dvě metriky totiž odchylku počítají zvlášť pro každou ze složek polohového vektoru bodu zvlášť. Důsledkem toho bude, že odchylka $d = \Delta x + \Delta y \approx 1.41$, tedy vyšší, než by měla být. Experimentální výsledky metrik pro dvě shodné animace, jen jinak natočené uvádí tabulka 3.1.



Obrázek 3.6: Invariance metriky vůči natočení modelu: a) Bod $p \in \mathcal{M}$ a jeho modifikovaná verze p' vzniklá posunem o $d = 1$ ve směru osy x . b) Dvojice bodů p a p' po otočení modelu o $\frac{\pi}{4}$. Pokud metrika počítá odchylku po složkách, dojde v obou případech k jinému výsledku.

natočení	0°	45°	dif(%)
MSE (10^3)	1.60	1.60	0.0
HD (10^3)	3.63	3.59	-1.2
HD RMS (10^3)	0.74	0.74	-0.2
KG	0,24	0,47	90.9
VCHV (10^3)	1.02	1.02	0.0
VCHVs (10^3)	1.62	1.63	0.3
TD (10^3)	0.20	0.20	0.5
4D HD (10^3)	0.44	0.60	35.9
RibbonA (10^3)	0.50	0.61	21.1
RibbonP (10^3)	3.55	4.26	19.9

Tabulka 3.1: Porovnání výsledných hodnot jednotlivých metrik pro animaci *hranolu* s konstantní distorzí po otočení modelu o 45°. Odchytky jednotlivých metod pro obě animace uvádí poslední řádek v procentech. Původní odchytky metrik (kromě KG) byly pro lepší zobrazení vynásobeny 10^3 .

3.3.2 Změna měřítka

Vezměme si opět náš model hranolu a představme si, že ho z nějakého důvodu potřebujeme zvětšit, tedy změnit měřítko, ale na topologii ani metrikách objektu se nic nezmění. V takovém případě bychom jistě očekávali, že pro stejnou míru distorze dostaneme od srovnávacích metod pro oba modely stejné, nebo alespoň podobné hodnoty, tedy hodnoty nezávislé na zvětšení či zmenšení modelu. Z tohoto důvodu se u většiny metod z kap.2 zavádí parametr, který takovou invarianci zajistí. Většinou se jedná o nějakou globální míru dynamické sítě, jako je např. hlavní diagonála. Touto konstantou poté dělíme vypočtené odchylky a dostáváme tak hodnoty nezávislé na měřítku modelu. V tomto ohledu nemají popsané metriky problémy a podávají očekávané výsledky, jak ukazují i výsledky experimentálního testování, které zobrazuje tabulka 3.2.

zvětšení	1x	2x	dif(%)
MSE (10^2)	3.60	3.60	0.0
HD (10^2)	4.48	4.48	0.0
HD RMS (10^2)	0.51	0.51	0.0
KG	13.29	13.29	0.0
VCHV (10^4)	0.80	0.80	0.0
VCHVs (10^4)	5.98	5.98	0.0
TD (10^7)	1.16	1.16	0.0
4D HD (10^3)	0.99	0.99	0.0
RibbonA (10^2)	1.15	1.15	0.0
RibbonP (10^2)	0.50	0.50	0.0

Tabulka 3.2: Porovnání výsledných hodnot jednotlivých metrik pro animaci *hranolu* s konstantní distorzí pro dvojnásobnou změnu měřítka modelu. Odchylky jednotlivých metod pro obě animace uvádí poslední řádek v procentech.

3.3.3 Počet snímků

Podobně jako v předchozím odstavci, také délka animace by neměla výrazně ovlivnit výsledky metriky, ačkoliv se to tak na první pohled jevit nemusí. Představme si ale náš model hranolu, který se pohybuje stále jedním směrem a v čase má kon-

stantní distorzi, např. je mírně prohnutý. Pro pozorovatele pak nebude velký rozdíl, zda animace bude trvat 10, nebo 20 sekund. Vždy uvidí ten stejný hranol v pohybu. A stejně by situaci měla chápat i objektivní metrika. Řešení je podobné jako v případě invariance vůči měřítku. Tentokrát se ale jako relativní míra bere délka animace, neboli počet snímků. Experimentální hodnoty metrik pro dvě stejné, jen jinak dlouhé animace uvádí tabulka 3.3.

poč.snímků	18	36	dif(%)
MSE (10^3)	4.01	4.01	0.0
HD (10^3)	6.24	6.24	0.0
HD RMS (10^3)	2.98	2.98	0.0
KG	1.53	1.54	0.5
VCHV (10^7)	4.84	5.34	10.4
VCHVs (10^7)	9.70	11.55	19.0
TD (10^8)	5.76	5.76	0.0
4D HD (10^3)	2.37	2.37	0.0
RibbonA (10^2)	1.29	1.32	2.5
RibbonP (10^2)	7.04	7.04	0.0

Tabulka 3.3: Porovnání výsledných hodnot jednotlivých metrik pro animaci *hranolu* s konstantní distorzí pro dvojnásobnou délku animace. Odchyly jednotlivých metod pro obě animace uvádí poslední řádek v procentech.

3.3.4 Počet bodů

Nezávislost výstupů metrik na počtu bodů sítě by se mohla na první pohled zdát zvláštní, jelikož pokud máme jiný počet bodů, jedná se v podstatě o jiný model. Představme si však situaci, že máme např. model koule, která je popsána parametricky. Definicí počtu svislých a podélných řezů můžeme určovat “hladkost” povrchu. Máme tedy modely stejného objektu, ale s různou hustotou vzorkování a tedy různým počtem bodů sítě jako vstupu pro porovnávací metriky. Bez ohledu na to je zřejmě oprávněné očekávat podobné hodnoty pro alespoň blízké frekvence vzorkování. Metriky, které jsme uvedli, s takovou situací přímo nepočítají. Provedli jsme experimentální měření na parametrickém modelu koule pro různý počet řezů a vý-

sledky zobrazuje tabulka 3.4. Je vidět, že až na chybové vektory a trojúhelníkovou diferencí se se situací metriky vyrovnaly celkem obstojně.

poč. řezů	8	12	16	20	24	dif(%)
MSE (10^3)	3.56	3.62	3.65	3.67	3.69	1.1
HD (10^2)	2.15	2.05	1.99	1.96	1.95	3.1
HD RMS (10^3)	9.98	9.78	9.68	9.63	9.61	1.2
KG	3.13	2.94	2.88	2.85	2.84	2.9
VCHV (10^4)	8.29	5.00	3.53	2.72	2.21	42.2
VCHVs (10^3)	9.17	8.05	7.90	7.94	7.61	5.1
TD (10^3)	3.78	1.37	0.68	0.40	0.27	78.3
4D HD (10^4)	5.72	5.47	4.99	4.97	4.93	5.8
RibbonA (10^3)	3.24	3.33	3.37	3.39	3.40	1.5

Tabulka 3.4: Porovnání výsledných hodnot jednotlivých metrik pro animaci *koule* s konstantní distorzí pro různou hustotu vzorkování. Hodnota odchylky (%) udává průměrnou odchylku hodnot od jejich střední hodnoty v procentech.

Kapitola 4

Výsledky objektivních metod

4.1 Implementace

Všechny metody zmíněné v kapitole 2 byly realizovány jako samostatné moduly pro systém *MVE-2 (Modular Virtual Environment 2)*, který je vyvíjen v rámci *Centra počítačové grafiky a vizualizace dat* katedry informatiky na Západočeské univerzitě v Plzni. Implementace metrik *MSE*, *KG*, *TD* a *Ribbon* je přímočará a není třeba ji zde detailně popisovat.

Řešením metody založené na Hausdorffově vzdálenosti se již zabývali Aspert a kol. [Aspe02] a také Cignony a kol. [Cign96]. Výsledkem druhé ze zmíněných prací byl nástroj na vyhodnocení podobnosti dynamických sítí pod názvem *METRO*. Tento program je volně šiřitelný ve formě knihovny. *MVE* modul použitý v této práci pro měření podobnosti Hausdorffovou vzdáleností je na tomto nástroji založen a pouze volá jeho metody pro vstupní dynamické sítě a vrací hodnotu metriky jako jejich srovnání. Pro metriku 4D Hausdorffovy vzdálenosti byla využita implementace Libora Váši [VaSk06].

Pokud jde o metodu chybových vektorů, pak při implementaci s využitím topologické sousednosti bodů nenarazíme na větší problém. Horší je situace v případě, že chceme vyřešit problém uvedený na obr.3.2 a místo topologických sousedů budeme uvažovat sousedy prostorové. Za takové sousedy daného bodu budeme považovat všechny body modelu, které se nachází uvnitř kulového okolí, definovaném jeho po-

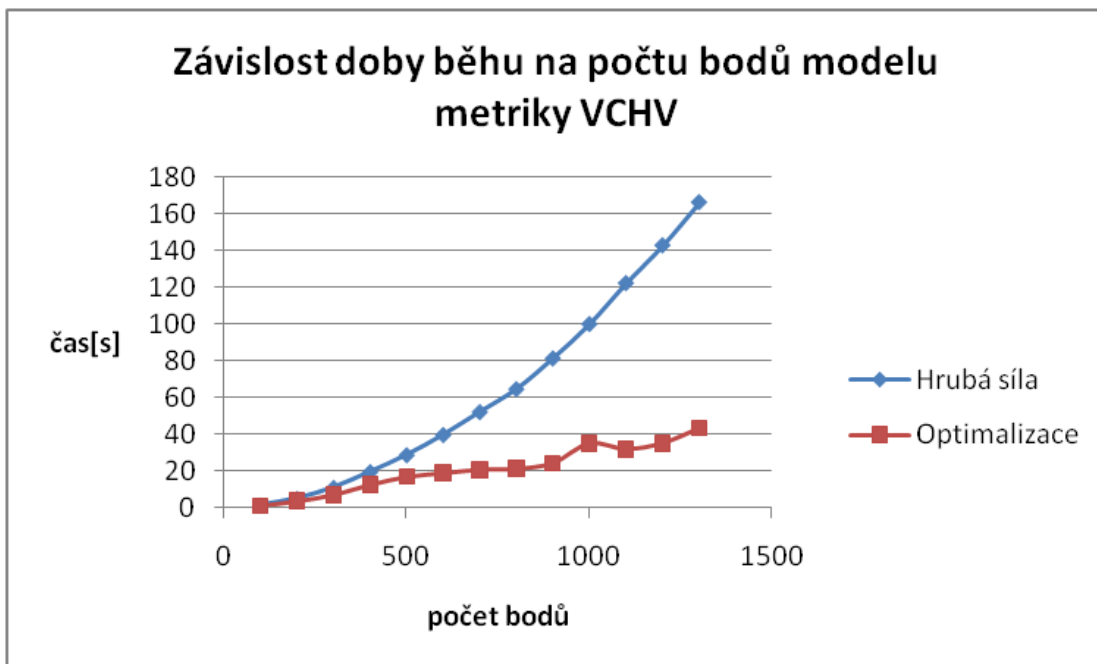
loměrem. Při naivní implementaci bychom dospěli pro každý snímek animace ke složitosti $O(N^2)$, kde N je počet bodů modelu. Takový výpočet by byl pro komplexnější modely, či delší animace z časových důvodů nepoužitelný. Při experimentech s použitím modelu *chicken*, který má přibližně 3.500 bodů se již pro 200 snímků pohybovala doba běhu algoritmu v řádech hodin. Při implementaci modulu této metody jsme proto využili prostorového dělení. Model jsme rozdělili do pravidelné trojrozměrné mřížky. Při hledání sousedních bodů pak algoritmus prohledává pouze ty buňky mřížky, které leží nebo jsou alespoň protnuté okolím bodu. Odstraníme tak zbytečné prohledávání všech bodů modelu. Pro srovnání jsme implementovali ve formě modulů oba přístupy, jak naivní, tak optimalizovaný. Graf porovnání dosažených časů obou algoritmů v závislosti na počtu bodů modelu je uveden na obr.4.1. Výstupní hodnoty metriky pro optimalizovanou variantu byly verifikovány vůči výsledkům metody hrubé síly s přesností na 10^{-9} .

Metoda chybových vektorů má ještě jeden důležitý aspekt a tou je časová vzdálenost, do které se nahlíží při výpočtu průměrného chybového vektoru okolí daného bodu modelu. Tedy kolik předchozích a následujících snímků bude do časového okolí zahrnuto. Pro snížení paměťových nároků bylo při implementaci použito plovoucí okno, ve kterém jsou uloženy vždy jen hodnoty chybových vektorů pro snímky, které jsou právě potřeba k výpočtu.

Detailní dokumentace modulů *MVE* pro všechny metody a popis jejich použití je uveden v příloze B.

4.2 Moduly distorze

Aby bylo možné v průběhu testování nasimulovat některé typy artefaktů, byly naimplementovány další dva *MVE* moduly. Tyto přidávají do původní dynamické sítě určité narušení, přičemž je možno regulovat jeho míru pomocí parametrů modulu.



Obrázek 4.1: Experimentální výsledky závislosti doby běhu na počtu bodů modelu vstupní dynamické sítě pro metodu chybových vektorů. Měření bylo provedeno na vstupní datové sadě 'chicken' při délce 50 snímků.

4.2.1 Gaussovský šum

Tento modul vnáší do původní animace gaussovský šum. Ten se generuje pro každý další snímek zvlášť, takže není po celou dobu stálý a kromě nerovností povrchu lze tak simulovat kmitání bodů povrchu v čase. Vstupními parametry modulu jsou střední hodnota a rozptyl normálního rozložení náhodné veličiny. Jestliže se dále v textu budeme odkazovat (např. při testech) na šum v síti, budeme používat zkrácené označení $G(x)$, kde x je rozptyl (střední hodnota je vždy 0).

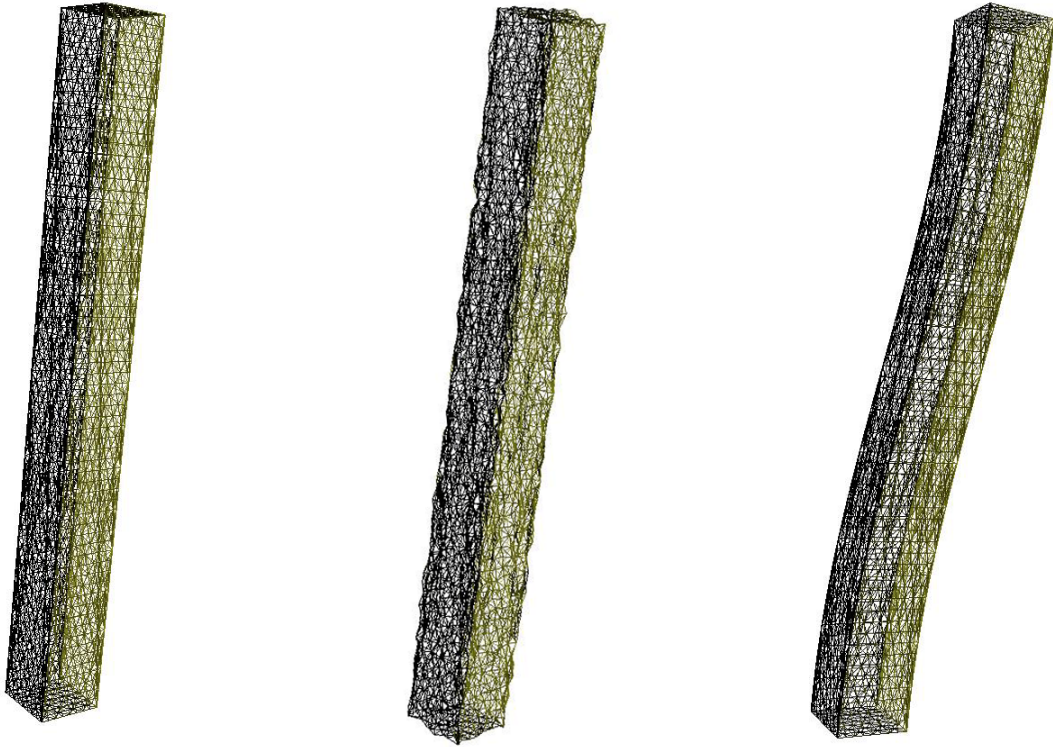
Při realizaci generátoru náhodných čísel s normálním (gaussovským) rozdělením jsme využili tzv. *metodu inverzní transformace*. Vyjdeme z generátoru, který produkuje sekvenci čísel s rovnoměrným rozdělením, tedy čísla z intervalu $\langle 0.0, 1.0 \rangle$ (v prostředí .NET Framework např. metoda `Random.nextDouble`). Střední hodnota takové řady je $\mu = 0.5$ a rozptyl $\sigma^2 = \frac{1}{12}$. Z uvedených charakteristik je zřejmé, že sečtením vždy dvanácti po sobě vygenerovaných čísel dostaneme řadu s normálním rozdělením $N(\mu, \sigma^2) = N(6, 1)$. Chceme-li získat rozdělení s libovolnými parametry střední hodnoty a rozptylu, stačí pouze transformace intervalu, tedy předpis gaussovského generátoru bude následující:

$$x = \mu + \sigma^2 \left[\left(\sum_{i=1}^{12} y_i \right) - 6 \right]. \quad (4.1)$$

4.2.2 Sinusové prohnutí

Užitečná pro testování je také simulace ohnutí nebo posunu některé části modelu. Proto druhý modul zavádí distorzi ve formě prohnutí modelu ve tvaru sinusové křivky. Vstupní parametry modulu definují plochu (xy, yz, zx) a řídicí osu (x, y, z) , které určují směr prohnutí. Definiční obor je definován intervalem $\langle \frac{\pi}{2}, \frac{3\pi}{2} \rangle$. Míra prohnutí je vyjádřena procenty vzhledem k rozměru modelu ve směru vedlejší osy prohnutí. Dále v textu budeme pro označení sinusového prohnutí používat zkratku $S(x)$, kde x představuje parametr funkce sinus vyjádřen v procentech rozměru modelu.

Ukázky výstupu obou modulů pro model hranolu jsou uvedeny na obr. 4.2.



Obrázek 4.2: Ukázka výstupu distorzních modulů. Jako originál byl použit model hranolu (vlevo). Na něj byl aplikována nejprve Gaussovský šum s parametry $\mu = 0, \sigma^2 = 0.2\%$ délky hlavní diagonály modelu (uprostřed) a sinusové ohnutí o velikosti 50% délky hrany podstavy.

4.3 Srovnání metrik

Dříve, než budeme jednotlivé metriky porovnávat z hlediska subjektivního vnímání, bylo by vhodné porovnat hodnoty, které metriky podávají, navzájem mezi sebou. Půjde nám o to zjistit, zda jsou hodnoty metrik na sobě nějak závislé (např. jen násobek), nebo mají zcela jiný průběh.

V teorii pravděpodobnosti a statistice se používá tzv. *korelace* (někdy také *koeficient korelace*) k určení míry lineární závislosti dvou náhodných veličin. Existuje několik různých koeficientů, které tuto míru definují. Nejznámější je Pearsonův korelační koeficient, který se počítá jako podíl očekávaných hodnot veličin a jejich rozptylů:

$$\rho_{X,Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}, \quad (4.2)$$

kde E je očekávaná hodnota. Korelace je definována pouze v případě, že rozptyly obou veličin jsou konečné a nenulové. Korelace je vždy v intervalu $\langle -1, 1 \rangle$, přičemž pokud $\rho_{X,Y} = 1$, mezi oběma veličinami existuje stoupající lineární závislost, při $\rho_{X,Y} = -1$ je lineární závislost klesající a pro $\rho_{X,Y} = 0$ jsou veličiny nezávislé. Všechny hodnoty mezi těmito hraničními označují stupeň závislosti.

Hodnoty, které dostaneme jako výstup metrik můžeme také považovat za náhodné veličiny a použijeme tedy techniku korelace pro srovnání jejich případné závislosti. Pokud máme více než dvě náhodné veličiny ke zpracování, pak se používá tzv. *korelační matice*, jejíž buňky udávají korelační koeficient vždy dvou z proměnných. Korelační matici pro naše metriky (byla použita animace *dance*) uvádí tabulka 4.1. Jak je vidět, metriky se rozdělují do dvou skupin. Metody MSE, HD, KG a Ribbon tvoří první z nich, přičemž mezi všemi jejími zástupci existuje téměř dokonalá lineární závislost. Stejně tak metriky VCHV a TD jsou dokonale lineárně závislé se vzrůstající tendencí. Jestliže porovnáme obě skupiny mezi sebou, i zde je výrazná lineární závislost, nicméně tentokrát je její trend klesající, tedy hodnoty se spíše blíží inverzním.

	MSE	HD	KG	VCHV	VCHV_s	TD	Ribbon
MSE	1.00						
HD	0.96	1.00					
KG	0.96	1.00	1.00				
VCHV	-0.49	-0.61	-0.60	1.00			
VCHV _s	-0.48	-0.60	-0.59	1.00	1.00		
TD	-0.42	-0.54	-0.53	1.00	1.00	1.00	
Ribbon	0.97	1.00	1.00	-0.60	-0.58	-0.52	1.00

Tabulka 4.1: Korelační matice pro objektivní metriky. Buňky matice uvádí vždy korelační koeficient mezi dvěma metrikami.

Kapitola 5

Subjektivní srovnání metrik

Jak jsme již zmínili, všechny uvedené metriky nám poskytují exaktní popis toho, nakolik jsou dva povrchy rozdílné, avšak pro pozorovatele se může leckdy situace jevit jinak, jelikož lidské oko je různě citlivé na různé druhy odchylek. Objektivní metody pro měření odchylek dynamických sítí by se tedy měly alespoň snažit kopírovat subjektivní pohled člověka, i když s použitím objektivních měřítek (více viz. kapitola 3). V této části práce se zaměříme právě na to, jak jsou defekty na původních sítích vnímány lidmi.

5.1 Metodika výzkumu

Pro ten účel byl vypracován výzkum, který měl za úkol odhalit, nakolik opravdu objektivní metody respektují zmíněná pravidla (3.2) pro kvalitní metriku. Šetření bylo prováděno osobně s využitím počítače (CAPI - Computer Aided Personal Interviewing). Do subjektivního testování bylo zahrnuto 15 respondentů. Všichni vyhodnocení prováděli za stejných podmínek a se shodným vybavením. Každý dostal soubor animací, přičemž byl vždy označen jen originál a deformované verze byly řazeny náhodně. Své odpovědi respondenti zaznamenávali do dotazníků (ukázka dotazníku viz. příloha A). Otázky byly zaměřeny na rozpoznatelnost různých artefaktů a míru jejich rušivosti při sledování sekvencí.

Pro jednodušší manipulaci byly všechny animace vygenerovány jako VRML soubory a k jejich prohlížení jsme použili volně šiřitelné prohlížeče *Cortona VRML*

*Client*¹ a *Flux Player*².

5.2 Relevantnost výsledků

U každého průzkumu je důležitá otázka relevantnosti dosažených výsledků. Potřebujeme stanovit míru, do jaké lze brát získaná data za spolehlivá. Velmi často se pro tyto účely používá statistický nástroj známý jako *interval spolehlivosti*. Ten vlastně vymezuje rozsah, ve kterém můžeme hodnoty sledovaného parametru se zvolenou pravděpodobností očekávat.

Důležitým parametrem při výpočtu intervalu spolehlivosti je α , který bývá označován jako *hladina významnosti*. Z něj je poté odvozena *hladina spolehlivosti* jako $100(1 - \alpha)\%$. Pro koeficient α se většinou volí hodnoty 0.05, nebo 0.01, tedy hladina spolehlivosti je potom 95%, resp. 99%. Výsledný interval pak označujeme jako $100(1 - \alpha)\%$ interval spolehlivosti a zapíše se jako $E(x) \pm i$, kde E označuje střední hodnotu náhodné proměnné x a i je rovno polovině velikosti intervalu spolehlivosti.

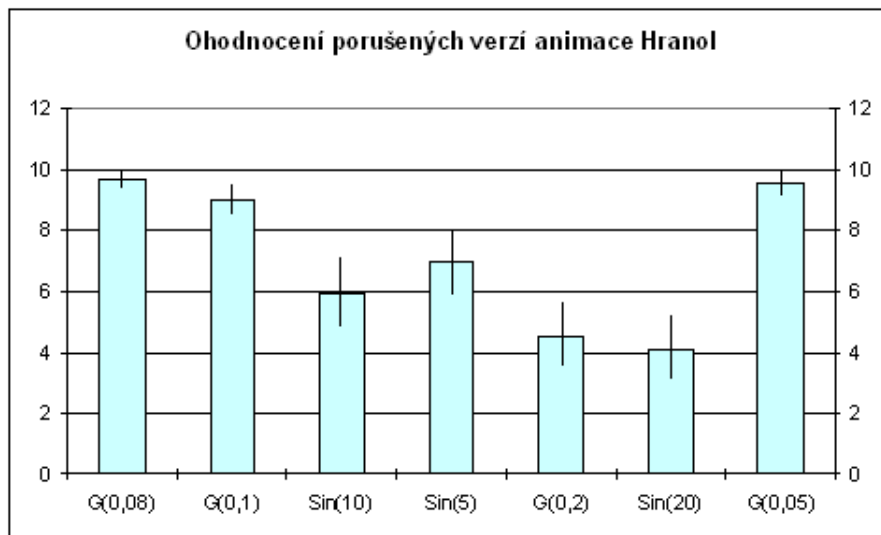
5.3 Vyhodnocení

Pro označení průměrného hodnocení ukázky se v průzkumech často používá označení MOS (Mean Opinion Score), což je prostý aritmetický průměr známek od všech respondentů zúčastněných na testování. Užívána je především pokud dotazovaný má ohodnotit kvalitu nějakého jevu na stupnici (např. od 1 do 10, což je náš případ). My se přidržíme tohoto značení a budeme ho používat pro průměrnou známku degenerovaných variant animací, tak jak je účastníci hodnotili a stejně tak pro hodnotu rušivosti artefaktů, které se v datech vyskytly.

Jak již bylo řečeno, hlavním cílem testování bylo zjištění, do jaké míry objektivní metriky kopírují subjektivní vjem pozorovatele. Abychom získané subjektivní i objektivní hodnocení mohli nějak porovnat, bylo nutné tyto znormalizovat, takže

¹Cortona VRML Client - <http://www.parallelgraphics.com/products/cortona>

²Flux Player - <http://www.mediamachines.com>

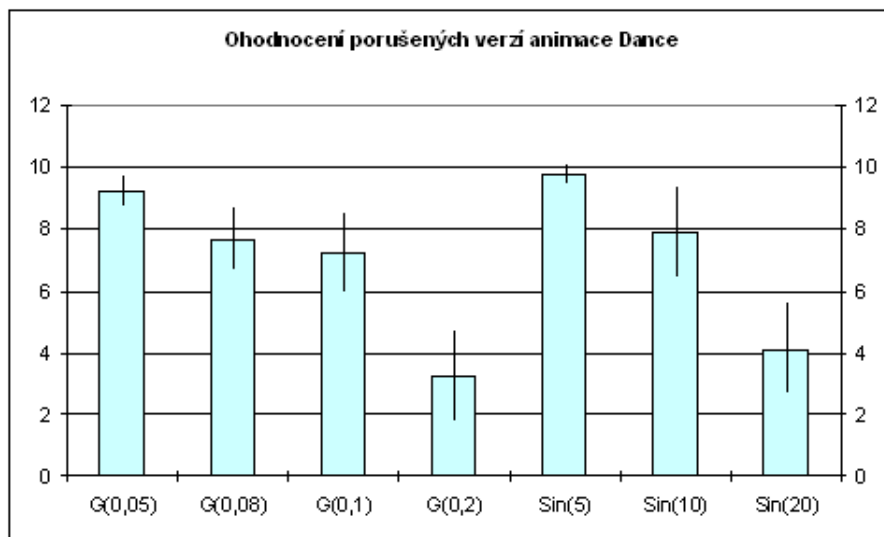


Obrázek 5.1: Průměrné známky sedmi modifikovaných verzí animace *hranol* s 95% intervalem spolehlivosti.

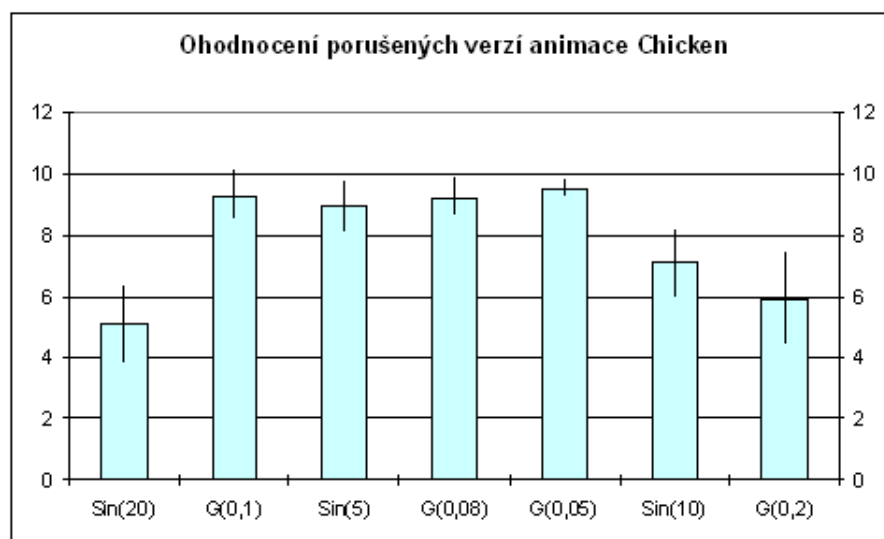
veškeré hodnoty dostaneme transformované na interval $\langle 0; 1 \rangle$. Dolní hranicí intervalu je vždy nula, tedy dvě porovnávané dynamické sítě jsou shodné, jako horní hranici jsme brali vždy ohodnocení té nejhorší verze sítě. Vzhledem k tomu, že se chceme co nejvíce objektivními metodami přiblížit subjektivním, vyjádřili jsme také průměrnou odchylku vůči hodnotám *MOS*.

Otázky dotazníku se zaměřovaly na ohodnocení kvality jednotlivých deformovaných variant sítě. Respondent měl označit na stupnici od 1 od 10 podobnost s originálem. Grafy 5.1, 5.2 a 5.3 uvádí průměrné známky včetně intervalů spolehlivosti pro použité dynamické sítě. Záměrně jsme pro testování vybrali tři modely, které jsou povahově různé. Model *hranol* je naprosto pravidelný a je ukázkou umělého objektu. *Dance* a *Chicken* naproti tomu představují reálné objekty, liší se však strukturou a rozměry.

Tabulka 5.1 uvádí normalizované hodnoty *MOS* pro datovou sadu *hranol* v porovnání s objektivními metrikami a průměrnými odchylkami jednotlivých metod v horní části, ve spodní části je pak uvedeno pořadí od nejlepší (1) po nejhorší (7) variantu sítě, tak jak je metody seřadily. Model *hranolu* byl vybrán záměrně jako objekt s pravidelnými rozměry a rovnými hranami, aby bylo možné porovnat, jak se



Obrázek 5.2: Průměrné známky sedmi modifikovaných verzí animace *dance* s 95% intervalem spolehlivosti.



Obrázek 5.3: Průměrné známky sedmi modifikovaných verzí animace *chicken* s 95% intervalem spolehlivosti.

metody zachovají v porovnání s ohodnocením realističtějších modelů, jako je např. *dance*. U takového pravidelného objektu totiž atrefakty mají zcela jiný dopad, než u reálných objektů. Např. jakékoliv ohnutí znamená velký problém pro pozorovatele, zatímco jiné defekty, jako např. nepatrné zhrbolatění povrchu bude spíše akceptovatelné. To potvrzují také výsledky subjektivního testování. Nejlepších výsledků zde dosahují metody MSE, KG či Ribbon, zatímco chybové vektory jsou na tom nejhůř, jelikož posunutí částí modelů nepovažuje tato metoda za závažný problém. Průměrná odchylka vzhledem k hodnotám MOS se u těchto metod blíží k 0.5, tedy téměř polovina celého intervalu. Do stejné kategorie zapadá metoda založená na diferencích obsahu trojúhelníků a není tedy stejně tak pro umělé modely tou nejlepší volbou.

V praxi ale většina dynamických sítí bude složitějších tvarů, než je hranol. Stejně testování jsme tedy provedli také pro animace *dance* a *chicken*. Výsledky porovnání všech metod se subjektivním hodnocením jsou uvedeny v tabulkách 5.2 a 5.3. Jak je vidět, zde je situace opačná oproti předchozímu. Metriky jsou opět rozděleny do dvou, potažmo tří, skupin. První, do které spadají metody MSE, HD, KG a Ribbon, tentokrát dopadá hůře, jelikož na realističtějších modelech pozorovatel ohnutí či posuny zdaleka nevnímá tak citlivě, zatímco narušení hladkosti povrchu či kmitavý pohyb je nepřijatelný. O mnoho lépe tedy o kvalitě deformovaných verzí sítí vypovídají metody chybových vektorů. Zcela nejlepšího výsledku pak dosáhla metoda TD s průměrnou odchylkou 0.188 od hodnoty MOS. V těchto měřeních již není zahrnuta metrika 4D HD, jelikož vzhledem k paměťovým nárokům není pro tyto modely použitelná.

Ukázali jsme si tedy srovnání metrik se subjektivním vjemem člověka. Pro lepší představu jak vlastně porovnání zkusíme opět použít korelaci, tentokrát mezi metrikami a hodnotami MOS. Tabulka 5.4 zobrazuje tyto hodnoty postupně pro datové sady *hranol*, *dance* a *chicken*. Z uvedených výsledků je zřejmé, že metriky podávají velmi rozdílné výsledky v závislosti na charakteru vstupních dat. Pro umělý objekt hranolu a model *chicken* vyznívá situace výrazně lépe pro metriky MSE, HD, KG a Ribbon. Naopak u lidské postavy v datové sadě *dance* se výsledky obrací a tentokrát vychází nejlépe metoda TD.

metrika	G(0.08)	G(0.05)	G(0.1)	S(5)	S(10)	G(0.2)	S(20)	odch.
MOS	0.050	0.083	0.165	0.579	0.777	0.975	1.000	-
MSE	0.006	0.002	0.010	0.063	0.250	0.038	1.000	0.323
HD	0.052	0.033	0.064	0.258	0.510	0.121	1.000	0.228
KG	0.078	0.049	0.098	0.250	0.500	0.195	1.000	0.216
VCHV	0.400	0.250	0.500	0.000	0.000	1.000	0.001	0.462
VCHVs	0.401	0.250	0.500	0.000	0.001	1.000	0.001	0.462
TD	0.394	0.248	0.493	0.000	0.001	1.000	0.002	0.459
4D HD	0.052	0.035	0.062	0.241	0.489	0.095	1.000	0.237
Ribbon	0.097	0.061	0.122	0.250	0.500	0.243	1.000	0.207
MOS	1	2	3	4	5	6	7	
MSE	2	1	3	5	6	4	7	
HD	2	1	3	5	6	4	7	
KG	2	1	3	5	6	4	7	
VCHV	5	4	6	1	2	7	3	
VCHVs	5	4	6	1	2	7	3	
TD	5	4	6	1	2	7	3	
4D HD	2	1	3	5	6	4	7	
Ribbon	2	1	3	5	6	4	7	

Tabulka 5.1: Porovnání výsledných hodnot jednotlivých metrik pro animaci *hranol*. $G(x)$ označuje gaussovský šum o střední hodnotě rozložení 0 a rozptylu x , $S(y)$ je pak sinusové prohnutí modelu s amplitudou y .

metrika	S(5)	G(0.05)	S(10)	G(0.08)	G(0.1)	S(20)	G(0.2)	odch.
MOS	0.091	0.121	0.348	0.364	0.439	0.864	1.000	-
MSE	0.063	0.000	0.250	0.000	0.001	1.000	0.003	0.312
HD	0.255	0.011	0.492	0.018	0.022	1.000	0.041	0.325
KG	0.250	0.014	0.500	0.022	0.028	1.000	0.055	0.322
VCHV	0.001	0.250	0.001	0.400	0.500	0.002	1.000	0.218
VCHVs	0.006	0.250	0.012	0.400	0.499	0.023	1.000	0.213
TD	0.020	0.233	0.039	0.372	0.468	0.079	1.000	0.188
4D HD	-	-	-	-	-	-	-	-
Ribbon	0.250	0.017	0.500	0.027	0.034	1.000	0.068	0.318
MOS	1	2	3	4	5	6	7	
MSE	5	1	6	2	3	7	4	
HD	5	1	6	2	3	7	4	
KG	5	1	6	2	3	7	4	
VCHV	1	4	2	5	6	3	7	
VCHVs	1	4	2	5	6	3	7	
TD	1	4	2	5	6	3	7	
4D HD	-	-	-	-	-	-	-	
Ribbon	5	1	6	2	3	7	4	

Tabulka 5.2: Porovnání výsledných hodnot jednotlivých metrik pro animaci *dance*. $G(x)$ označuje gaussovský šum o střední hodnotě rozložení 0 a rozptylu x , $S(y)$ je pak sinusové prohnutí modelu s amplitudou y .

metrika	G(0.08)	G(0.05)	G(0.1)	S(5)	S(10)	G(0.2)	S(20)	odch.
MOS	0.102	0.163	0.143	0.224	0.592	0.837	1.000	-
MSE	0.000	0.000	0.000	0.063	0.250	0.002	1.000	0.249
HD	0.026	0.017	0.033	0.265	0.521	0.064	1.000	0.174
KG	0.018	0.011	0.022	0.250	0.500	0.044	1.000	0.181
VCHV	0.400	0.250	0.500	0.005	0.010	1.000	0.020	0.384
VCHVs	0.400	0.250	0.500	0.202	0.404	1.000	0.807	0.187
TD	0.157	0.061	0.246	0.017	0.070	1.000	0.307	0.248
4D HD	-	-	-	-	-	-	-	-
Ribbon	0.022	0.014	0.028	0.250	0.500	0.056	1.000	0.178
MOS	1	2	3	4	5	6	7	
MSE	2	1	3	5	6	4	7	
HD	2	1	3	5	6	4	7	
KG	2	1	3	5	6	4	7	
VCHV	5	4	6	1	2	7	3	
VCHVs	3	2	5	1	4	7	6	
TD	4	2	5	1	3	7	6	
4D HD	-	-	-	-	-	-	-	
Ribbon	2	1	3	5	6	4	7	

Tabulka 5.3: Porovnání výsledných hodnot jednotlivých metrik pro animaci *chicken*. $G(x)$ označuje gaussovský šum o střední hodnotě rozložení 0 a rozptylu x , $S(y)$ je pak sinusové prohnutí modelu s amplitudou y .

model	hranol	dance	chicken
MSE	0.61	0.46	0.72
HD	0.70	0.35	0.72
KG	0.73	0.36	0.71
VCHV	-0.02	0.53	0.09
VCHVs	-0.02	0.54	0.82
TD	-0.01	0.60	0.59
4D HD	0.68	-	-
RibbonA	0.75	0.37	0.72

Tabulka 5.4: Hodnoty korelačního koeficientu pro soubory objektivních hodnot metrik vzhledem k subjektivní metrice MOS.

Kapitola 6

Závěr

6.1 Zhodnocení výsledků a budoucí práce

Práce splnila všechny určené požadavky. Přinesla souhrn v současnosti známých metod na měření odchylek mezi dvěma dynamickými sítěmi a také přinesla návrhy na nové postupy v podání metod *chybových vektorů* a *trojúhelníkové difference*. Jak je vidět z měřených výsledků, není snadné jednoznačně určit, jaká metoda je pro měření optimální. Povahy výsledků je závislá na charakteru vstupních dat a je tedy i k tomuto potřeba vždy přihlédnout.

Rozšíření, která by mohla na tuto práci navazovat by se mohla týkat detailnějšímu srovnání subjektivního názoru pozorovatele, např. v rámci rozsáhlejšího testování. Dále se lze věnovat vylepšením konkrétních metod tak, aby došlo k odstranění ale-
spoň některých nežádoucích jevů, které byly v práci popsány.

Literatura

- [Anua04] Anuar, N., Guskov, I., *Extracting Animated Meshes with Adaptive Motion Estimation*. (Proceedings of the 9th International Fall Workshop on Vision, Modeling, and Visualization, November 2004).
- [Aspe02] Aspert, N., Santa-Cruz, D., Ebrahimi, T., *MESH: Measuring errors between surfaces using the Hausdorff distance*. (Proceedings of the IEEE International Conference on Multimedia and Expo, vol.I, p.705-708, 2002).
- [Cign96] Cignoni, P., Rochini, C., Scopigno, R., *METRO: Measuring error on simplified surfaces*. (Technical report B4-01-01-96, Istituto I.E.I. - C.N.R., Pisa, Italy, 1996).
- [Karn04] Karni, Z., Gotsman, C., *Compression of soft-body animation sequences*. (Center for Graphics and Geometric Computing, Faculty of Computer Science, Haifa, Israel, 2004).
- [Sand03] Sand, P., McMillan, L., Popovic, J., *Continuous Capture of Skin Deformation*. (ACM Transactions on Graphics. 22(3), pp. 578-586, 2003).
- [Vasa06] Váša, L., *Methods for dynamic mesh size reduction*. (Technical report DCSE/TR-2006-07).
- [VaSk06] Váša, L., Skala, V., *A spatio-temporal metric for dynamic mesh comparison*. (In proceedings of AMDO2006 International Conference, Spain, Springer-Verlag LNCS 4069, p.29-37, 2006). University of West Bohemia, Doctoral thesis, 2006).

Příloha A

Dotazník

Ukázka dotazníku, který byl předkládán respondentům průzkumu v rámci subjektivního testování vlivu různých deformací na vnímání pozorovatelem:

Příloha B

Uživatelský manuál

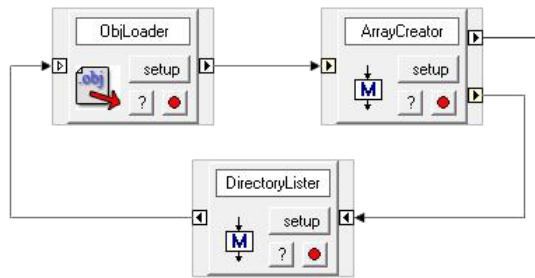
Jak bylo již zmíněno, veškeré implementované metody v této práci byly realizovány ve formě modulů pro systém MVE-2. V této části je popsáno, jak moduly použít pro načtení, zobrazení a porovnání dynamických trojúhelníkových sítí. Všechny zmíněné mapy jsou obsaženy na přiloženém CD (viz. kap. C).

B.1 Načtení datové sady

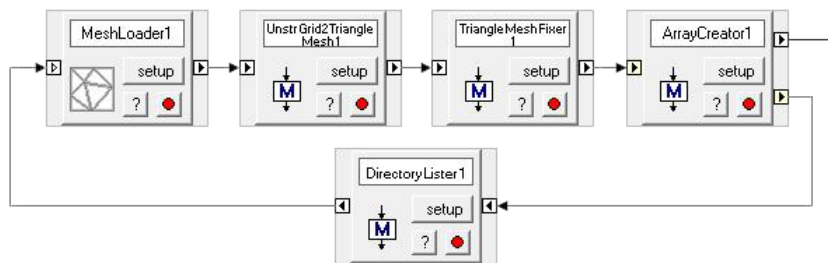
Datové sety použité v této práci byly dostupné ve formátu VRML, některé jako OBJ soubory, přičemž každý snímek je uložen ve vlastním souboru. Proto potřebujeme při načítání celé sekvence vytvořit smyčku, která bude procházet všechny soubory, postupně je načítat a ukládat do pole.

Jestliže máme tedy vstupní data ve formě OBJ souborů, pro jejich načtení použijeme smyčku modulů, jak ukazuje obrázek B.1. Zde modul `DirectoryLister` prochází zvolený adresář a vrací cesty souborů v něm obsažených. Další modul, `ObjLoader`, určené soubory načítá a předává modulu `ArrayCreator`. Ten načtený objekt zařadí do pole a předá modulu `DirectoryLister` index dalšího prvku v poli, který se má načíst. Konečný výstup, tedy kompletní dynamická síť se po zpracování objeví na výstupním portu modulu `ArrayCreator` a je připraven k libovolnému dalšímu zpracování (např. rendering, porovnání).

Pokud máme vstupní data ve formátu VRML, je mapa pro načtení o něco složitější. Modul `ObjLoader` nahradí `MeshLoader`, který však výsledek nepředá rovnou na



Obrázek B.1: Smyčka modulů pro načtení dynamické sítě v OBJ formátu.

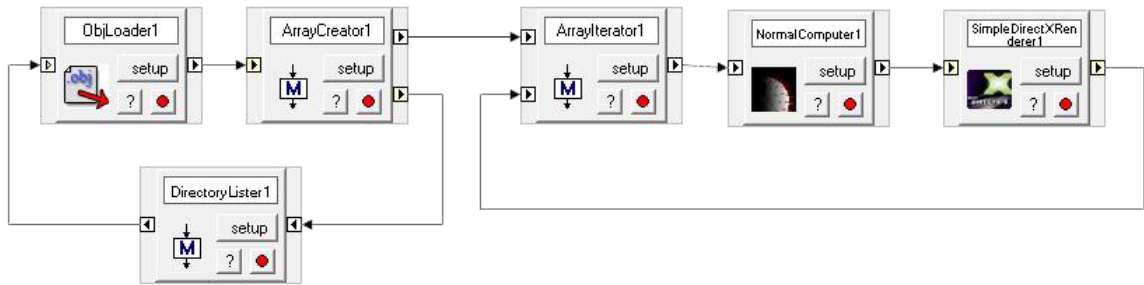


Obrázek B.2: Smyčka modulů pro načtení dynamické sítě ve VRML formátu.

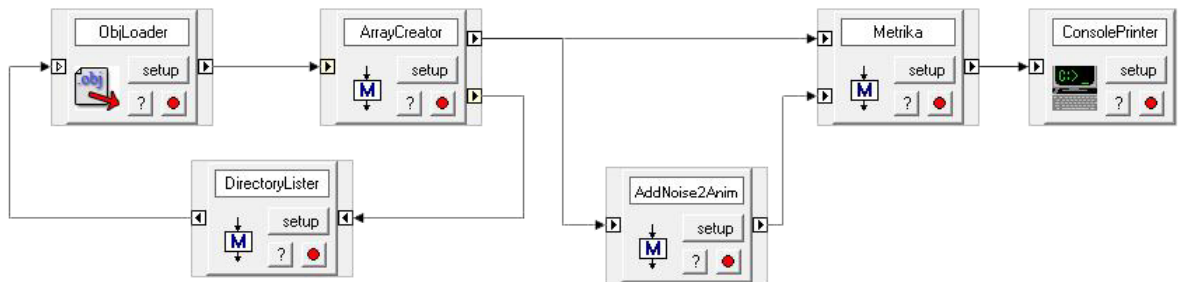
vstupní port ArrayCreatory. Výstup musí nejprve projít zpracováním dalšími dvěma moduly, kterými jsou UnstrGrid2TriangleMesh (převéde objekt na trojúhelníkovou síť) a TriangleMeshFixer (opraví případné chyby sítě). Mapa je na obr. B.2.

B.2 Vykreslení datové sady

Dalším krokem po načtení dat může být jejich zobrazení. Jelikož pracujeme s 3D modely, pro správné zobrazení potřebujeme pro každý snímek spočítat nejprve jeho normály. To má na starosti modul NormalComputer. Pro vykreslení jsme použili modul SimpleDirectXRenderer. Stejně jako u načítání, i tentokrát potřebujeme ale vytvořit smyčku, která projde všechny snímky animace. K tomu složí modul ArrayIterator, který podle indexu na vstupním portu posílá na výstup určitý snímek. Kompletní mapu načtení a vykreslení datové sady můžete vidět na obr. B.3.



Obrázek B.3: Smyčka pro vykreslení dynamické sítě.



Obrázek B.4: Smyčka pro měření odchyly dvou dynamických sítí.

B.3 Měření odchyly

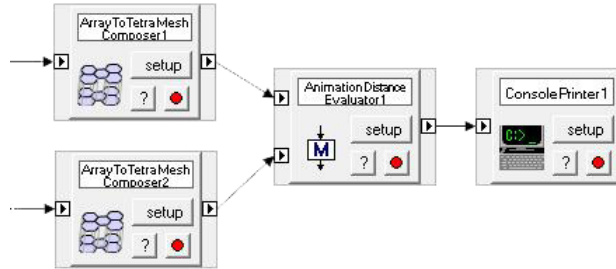
Hlavní náplní práce však je srovnání dvou dynamických sítí, takže si nyní ukážeme jak vytvořit mapu, která nám k tomuto účelu poslouží. Načítací smyčka bude stále stejná, jen pokud budeme mít k dispozici upravenou verzi sítě rovněž uloženou v souborech, pak budou smyčky v mapě dvě, pro každou ze sítí. My ale využijeme pro demonstraci jeden z implementovaných distorzních modulů (AddNoise2Anim, AddSinShift2Anim). Bude nám tedy stačit jediná smyčka pro načtení, přičemž její výsledek jednou použijeme v nezměněné formě a podruhé jej pošleme do distorzního modulu. Obecně všechny moduly pro porovnání dvou dynamických sítí mají dva vstupní porty, představující dvě vstupní sítě, a jeden výstupní pro hodnotu odchyly. Na obr. B.4 je vidět úplná mapa pro porovnání dvou sítí, kdy originál je narušen šumem. Výsledek metriky je ještě vypsán do konzole.

Následuje kompletní seznam modulů pro měření odchylek, které jsou k dispozici pro prostředí *MVE-2* v rámci projektu *Experiments*:

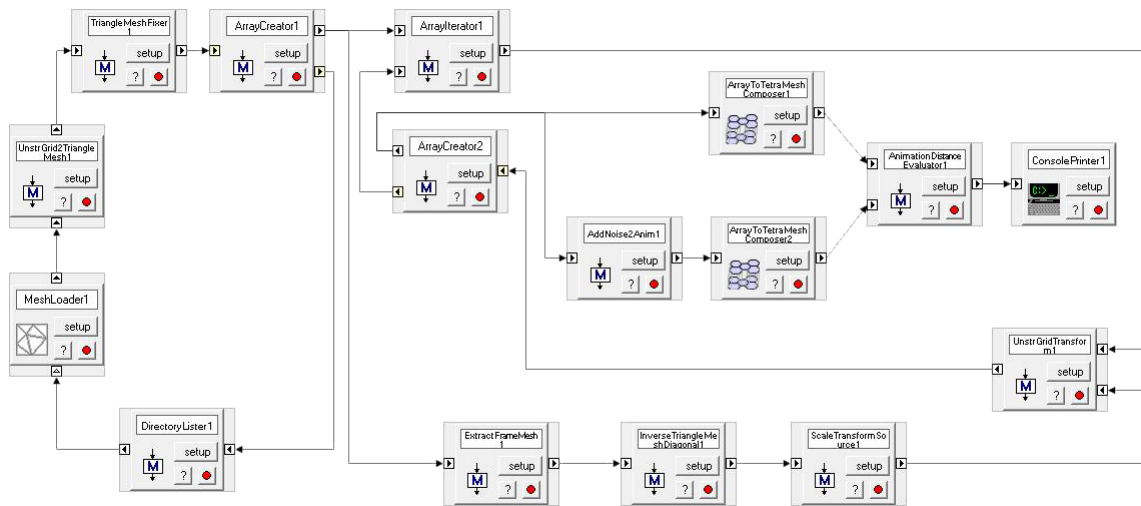
- AnimationComparerKG - metrika Kriege, Gotsman
- AnimationComparerMetro - metrika HD (pro použití modulu je potřeba soubor metro.exe, který je obsažen na přiloženém CD)
- AnimationComparerMSE - MSE
- AnimationComparerRibbon - Ribbon (parametr alfa nastavuje rozměr h (viz. kap.2.7))
- AnimationComparerTD - TD
- AnimationComparerVCHV - chybové vektory s topologickou sousedností
- AnimationComparerVCHVSpace - optimalizovaná verze chybových vektorů s prostorovou sousedností
- AnimationComparerVCHVsBrute - naivní verze chybových vektorů s prostorovou sousedností
- AnimationDistanceEvaluator - 4D HD

B.4 Použití 4D metriky

Jak bylo řečeno, předchozí postup lze použít pro většinu z uvedených srovnávacích metrik. Výjimkou je však metoda 4D Hausdorffovy vzdálenosti. Vzhledem ke specifické povaze této metody musíme také mapu pro její použití mírně upravit. Samotný porovnávací modul vyžaduje na vstupu místo klasických sekvencí trojúhelníkových sítí dvě tetrahedronové sítě. Proto musíme předřadit pro oba vstupy modul ArrayToTetraMeshComposer, který provede tento převod (obr. B.5).



Obrázek B.5: Zapojení modulu pro měření odchylky 4D Hausdorffovou vzdáleností.



Obrázek B.6: Kompletní zapojení modulu pro měření odchylky 4D Hausdorffovou vzdáleností včetně ošetření invariance vůči změně měřítka.

Příloha C

Obsah příloženého CD

Na příloženém CD naleznete v kořenovém adresáři soubor `main.htm`, který obsahuje přehlednou mapu celého disku a postup instalace a spuštění programu. V podadresáři `public` jsou umístěny binární soubory a knihovny potřebné pro správnou funkci systému *MVE-2*. Rovněž jsou k dispozici ukázková data a to v podadresáři `data`.